

(19)日本国特許庁 (JP)

(12) 公表特許公報 (A)

(11)特許出願公表番号

特表2003-515851

(P2003-515851A)

(43)公表日 平成15年5月7日(2003.5.7)

(51)Int.Cl.⁷

G 0 6 T 15/00
1/20

識別記号

1 0 0

F I

G 0 6 T 15/00
1/20

テ-マコード(参考)

1 0 0 A 5 B 0 5 7
C 5 B 0 8 0

審査請求 未請求 予備審査請求 有 (全127頁)

(21)出願番号 特願2001-542049(P2001-542049)
(86) (22)出願日 平成12年12月5日(2000.12.5)
(85)翻訳文提出日 平成14年6月6日(2002.6.6)
(86)国際出願番号 PCT/US00/33043
(87)国際公開番号 WO01/041069
(87)国際公開日 平成13年6月7日(2001.6.7)
(31)優先権主張番号 09/456,102
(32)優先日 平成11年12月6日(1999.12.6)
(33)優先権主張国 米国(US)

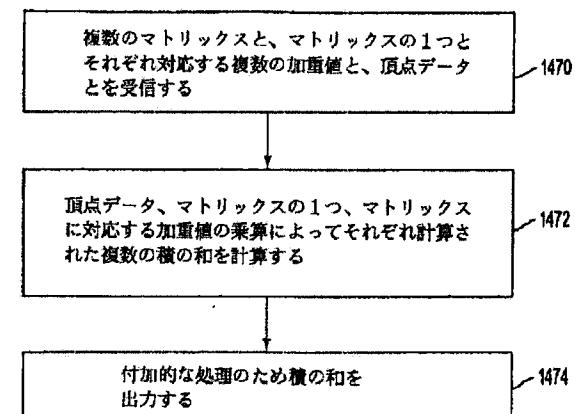
(71)出願人 エヌヴィディア
アメリカ合衆国、カリフォルニア州
95051 サンタ・クララ、モンロー・スト
リート 3535
(72)発明者 リンドホルム、ジョン
アメリカ合衆国、カリフォルニア州
95014 クバーティノ、バージニア・スワ
ン・プレイス 10325
(72)発明者 モイ、サイモン
アメリカ合衆国、カリフォルニア州
94040 マウンテン・ビュー、フリーダ
ム・レーン 100
(74)代理人 弁理士 鈴江 武彦 (外3名)

最終頁に続く

(54)【発明の名称】 グラフィックプロセッサ中の変換モジュール用の方法、装置および製品

(57)【要約】

ハードウェア実行グラフィックバイオラインにおけるグラフィック処理中にブレンディング動作を行う方法は、複数のマトリックス、複数の加重値および頂点データをバッファにおいて受取り(1470)、複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重の少なくとも1つとの乗算により計算されたものであり(1472)、付加的な処理のために積の和を出力する(1474)ステップを含んでいる。



【特許請求の範囲】

【請求項1】 (a) 頂点データをベクトルの形態で受取り、ベクトル頂点データに関してベクトル演算を行うためのベクトル演算モジュールと、
(b) ベクトル演算モジュールからのスカラー頂点データをベクトル頂点データに変換するための、ベクトル演算モジュールに結合された変換モジュールと、
(c) ベクトル演算モジュールの出力を記憶して、その出力をベクトル演算モジュールにフィードバックするための、ベクトル演算モジュールに結合されたレジスタとを具備しているグラフィック処理中にスカラーおよびベクトル成分を処理するためのシステム。

【請求項2】 ベクトル演算モジュールは乗算器および加算器の少なくとも一方を含んでいる請求項1記載のシステム。

【請求項3】 ゼロレイテンシーはレジスタをバイパスすることによって達成される請求項1記載のシステム。

【請求項4】 レジスタは、ベクトル頂点データを発生するベクトル成分書き込みマスクを含んでいる請求項3記載のシステム。

【請求項5】 さらに、ベクトル演算モジュールの出力に関してスカラー演算を実行し、それによって頂点データをスカラーの形態でレンダリングするための適合されたスカラー演算モジュールを備えている請求項1記載のシステム。

【請求項6】 スカラー演算には、逆数または逆平方根演算が含まれる請求項5記載のシステム。

【請求項7】 (a) 頂点データをベクトルの形態で受取り、
(b) ベクトル頂点データに関してベクトル演算を行い、
(c) ベクトル演算から結果的に得られたスカラー頂点データをベクトル頂点データに変換し、
(d) ベクトル演算の出力を記憶し、
(e) ベクトル演算の記憶された出力に関して付加的なベクトル演算を行うステップを含んでいるグラフィック処理中にスカラーおよびベクトル成分を処理する方法。

【請求項8】 ベクトル演算は乗算または加算演算を含んでいる請求項7記

載の方法。

【請求項9】 ベクトル演算はゼロレイテンシーでベクトル演算の出力に関して行われる請求項7記載の方法。

【請求項10】 ベクトル演算の出力はレジスタ装置に記憶され、ゼロレイテンシーはレジスタ装置をバイパスすることによって達成される請求項9記載の方法。

【請求項11】 ベクトル演算の出力に関してスカラー演算を実行し、それによって頂点データをスカラーの形態でレンダリングするステップをさらに含んでいる請求項7記載の方法。

【請求項12】 スカラー演算は逆数または逆平方根演算を含んでいる請求項11記載の方法。

【請求項13】 ベクトル演算の出力がベクトルの形態である場合、その出力からスカラー頂点データを抽出するステップをさらに含んでいる請求項11記載の方法。

【請求項14】 抽出はマルチプレクサによって行なわれる請求項13記載の方法。

【請求項15】 受取られた頂点データはマルチプレクサによって操作される請求項7記載の方法。

【請求項16】 (a) 頂点データをベクトルの形態で受取るためのコードセグメントと、

(b) ベクトル頂点データに関してベクトル演算を行うためのコードセグメントと、

(c) ベクトル演算から結果的に得られたスカラー頂点データをベクトル頂点データに変換するためのコードセグメントとを含み、

(d) ベクトル演算の出力を記憶し、

(e) ベクトル演算の記憶された出力に関して付加的なベクトル演算を行うグラフィック処理中にスカラーおよびベクトル成分を処理するためのコンピュータ読み出し可能な媒体上に支持されているコンピュータプログラム。

【請求項17】 ベクトル演算は乗算または加算演算を含んでいる請求項1

6記載のコンピュータプログラム。

【請求項18】 ベクトル演算はゼロレイテンシーでベクトル演算の出力について行われる請求項16記載のコンピュータプログラム。

【請求項19】 ベクトル演算の出力はレジスタ装置に記憶され、ゼロレイテンシーはレジスタ装置をバイパスすることによって達成される請求項18記載のコンピュータプログラム。

【請求項20】 さらに、ベクトル演算の出力に関してスカラー演算を実行し、それによって頂点データをスカラーの形態でレンダリングするコードセグメントを含んでいる請求項16記載のコンピュータプログラム。

【請求項21】 スカラー演算は逆数または逆平方根演算を含んでいる請求項20記載のコンピュータプログラム。

【請求項22】 さらに、ベクトル演算の出力がベクトルの形態である場合、その出力からスカラー頂点データを抽出するコードセグメントを含んでいる請求項20記載のコンピュータプログラム。

【請求項23】 抽出はマルチプレクサによって行なわれる請求項22記載のコンピュータプログラム。

【請求項24】 受取られた頂点データはマルチプレクサによって操作される請求項16記載のコンピュータプログラム。

【請求項25】 (a) 複数のマトリックス、複数の加重値、および頂点データをバッファにおいて受取り、

(b) 複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重の少なくとも1つとの乗算により計算されたものであり、

(c) 付加的な処理のために積の和を出力するステップを含んでいるハードウェア構成グラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行う方法。

【請求項26】 マトリックスはモデルビューマトリックスを含んでいる請求項25記載の方法。

【請求項27】 付加的な処理には、表示するための積の和と合成マトリックスとの乗算が含まれている請求項26記載の方法。

【請求項28】 付加的な処理には、ライティング動作が含まれている請求項26記載の方法。

【請求項29】 マトリックスは逆マトリックスを含み、頂点データは正規ベクトルを含んでいる請求項25記載の方法。

【請求項30】 バッファの出力に結合された第1の入力を有し、頂点データを受取る乗算論理装置と、この乗算論理装置の出力に結合された第1の入力を有する演算論理装置と、この演算論理装置の出力に結合された入力を有するレジスタ装置と、ならびに乗算論理装置および演算論理装置に結合されたメモリとを備えた単一の集積回路によって行われ、メモリには、頂点データを処理するときのための複数の定数および変数が記憶されている請求項25記載の方法。

【請求項31】 バッファの出力に結合された第1の入力を有する乗算論理装置と、この乗算論理装置の出力に結合された第1の入力を有する演算論理装置と、この演算論理装置の出力に結合された入力を有するレジスタ装置と、演算論理装置またはレジスタ装置の出力に結合された入力を備えた反転論理装置と、この反転論理装置の出力と乗算論理装置の第2の入力との間に結合された変換モジュールと、乗算論理装置および演算論理装置に結合されたメモリと、ならびに演算論理装置の出力に結合された出力コンバータとを備えた単一の集積回路によって行われ、反転論理装置は逆数または逆平方根演算を行い、変換モジュールはスカラー頂点データをベクトル調印データに変換するように構成されており、メモリには頂点データを処理するときに使用される複数の定数および変数が記憶されており、出力コンバータは処理された頂点データを出力するようにライティングモジュールに結合されている請求項25記載の方法。

【請求項32】 (a) 複数のマトリックス、複数の加重値、および頂点データを受取るためのバッファと、

(b) このバッファに結合された単一の集積回路とを備えており、この集積回路は複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重の少なくとも1つとの乗算により計算されたものであり、

(c) 付加的な処理のために単一の集積回路から積の和が outputされるグラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行うため

のシステム。

【請求項33】 マトリックスはモデルビューマトリックスを含んでいる請求項32記載のシステム。

【請求項34】 付加的な処理には、表示するための積の和と合成マトリックスとの乗算が含まれている請求項33記載のシステム。

【請求項35】 付加的な処理には、ライティング演算が含まれている請求項33記載のシステム。

【請求項36】 マトリックスは逆マトリックスを含み、頂点データは正規ベクトルを含んでいる請求項32記載のシステム。

【請求項37】 単一の集積回路は、バッファの出力に結合された第1の入力を有し、頂点データを受取る乗算論理装置と、この乗算論理装置の出力に結合された第1の入力を有する演算論理装置と、この演算論理装置の出力に結合された入力を有するレジスタ装置と、ならびに乗算論理装置および演算論理装置に結合されたメモリとを備えた单一の集積回路によって行われ、メモリには、頂点データを処理するときに使用される複数の定数および変数が記憶されている請求項32記載のシステム。

【請求項38】 単一の集積回路は、バッファの出力に結合された第1の入力を有する乗算論理装置と、この乗算論理装置の出力に結合された第1の入力を有する演算論理装置と、この演算論理装置の出力に結合された入力を有するレジスタ装置と、演算論理装置またはレジスタ装置の出力に結合された入力を備えた反転論理装置と、この反転論理装置の出力と乗算論理装置の第2の入力との間に結合された変換モジュールと、乗算論理装置および演算論理装置に結合されたメモリと、ならびに演算論理装置の出力に結合された出力コンバータとを備えており、反転論理装置は逆数または逆平方根演算を行い、変換モジュールはスカラー頂点データをベクトル調印データに変換するように構成されており、メモリには頂点データを処理するときに使用される複数の定数および変数が記憶されており、出力コンバータは処理された頂点データを出力するようにライティングモジュールに結合されている請求項32記載のシステム。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明は、一般に、グラフィックプロセッサに関し、とくに、グラフィックパイプラインシステムの変換モジュールに関する。

【0002】**【従来の技術】**

従来技術を表す図1は、パイプライン方式のグラフィック処理システムを構成する一般的な従来技術のシステムを示している。このシステムにおいて、データソース10はプリミティブを既定する拡張された頂点のストリームを発生する。これらの頂点は、記憶するために一時に1つづつ頂点メモリ13によってパイプライングラフィックシステム12に送られる。頂点メモリ13からパイプライングラフィックシステム12中に拡張された頂点が受取られると、頂点は変換モジュール14によって変換され、ライティングモジュール16によって照明され、さらに、ラスター化装置18によるレンダリングのためにクリップされて設定され、それによってレンダリングされたプリミティブを発生し、これが後で表示装置20上に表示される。

【0003】

動作中、変換モジュール14は、頂点をモデル座標で受取ってこれらの3次元頂点をそれらのモデル座標から、それらが最終的に表示される2次元ウインドウに変換するために使用することができる。変換を行うために、ビューポート、ビューマトリックス、ワールドマトリックス、投影マトリックス等の標準的な変換パラメータが使用されてもよい。

【0004】

同時に、上述したパラメータによって、幾何学的変換により、あるオブジェクトに関する別のオブジェクトの位置が表現され、回転され、クリップされ、そのサイズが定められると共に、3次元情景において見ている位置、方向および遠近が変更されることを可能にする。3次元頂点をそれらのモデル座標から、それらが表示される2次元ウインドウに変換する座標変換には、典型的に、移行、回

転およびスケーリングの1以上が必要である。

【0005】

従来技術の変換システムは典型的に、変換プロセス中に別々に発生されたスカラーおよびベクトル値を処理する。たとえば、位置属性、すなわち(X, Y, Z, W)は乗算器および、または加算器のようなベクトル演算子により処理され、それによってスカラー値をレンダリングしてもよい。スカラー演算子がこのようなスカラー値を処理してもよいが、それは典型的にベクトル演算子によって再度処理されることはない。グラフィックパイプライン処理中に処理された頂点データのスカラーおよびベクトル形態の処理を統合しようとする試みは現在まで行われていない。

【0006】

変換モジュール14によって処理されるさらに別のプロセスはブレンディング、すなわち“スキニング”である。スキニングとは、オブジェクト間の結合部をブレンディングすることによって分割された多角形オブジェクトに現実性を加えるプロセスのことである。従来技術の図1Aは、スキニングが行なわれる前および後の1対のオブジェクト22を示している。

【0007】

【発明が解決しようとする課題】

通常、スキニングプロセスはコンピュータプログラムおよび汎用プロセッサを使用して行なわれる。したがって、専用回路に関連した利益、すなわち、速度、効率等を得るためにハードウェア上でスキニングを実施しようと試みられたことがない。

【0008】

【課題を解決するための手段】

変換システムがグラフィック処理するための方法、装置および製品が提供される。システムには入力バッファが含まれており、この入力バッファは頂点データを頂点属性バッファから受取るためにこれに結合されるように構成されている。乗算論理装置は、入力バッファの出力に結合された第1の入力を有している。演算論理装置もまた設けられており、それは乗算論理装置の出力に結合された第1

の入力を有している。演算論理装置の出力にはレジスタ装置の入力が結合されている。

【0009】

反転論理装置もまた設けられており、それは演算論理装置の出力に結合された入力を有し、逆数または逆平方根演算を行う。1実施形態において、変換モジュールの反転論理装置においてヌルW属性値を処理する方法が提供される。W属性がヌルである場合、ゼロによる除算は役に立たない無限大値を発生させるので、ラスター化装置の設定モジュールがスクリーンスペースにおいてエッジ方程式を発生できないため、ヌルW属性値の処理はとくに重要なものとなる。使用において、頂点データを受取ると、変換モジュールの反転論理装置は頂点データのW属性の値を識別する。識別されたW属性値がヌルである場合、頂点データのW属性に関する除算演算は最小および最大指数にクランプされる。このクランプされた値をラスター化装置の設定モジュールがエッジ方程式を発生するために使用する。

【0010】

さらに、反転論理装置の出力と乗算論理装置の第2の入力との間に結合された変換モジュールが含まれている。使用において、変換モジュールはスカラー頂点データをベクトル頂点データに変換するように作用する。

【0011】

メモリは乗算論理装置および演算論理装置に結合される。メモリには、頂点データを処理するために入力バッファ、乗算論理装置、演算論理装置、レジスタ装置、反転論理装置および変換モジュールと共同して使用される複数の定数および変数が記憶されている。最後に、出力コンバータは演算論理装置の出力に結合され、処理された頂点データをライティングモジュールに出力するためにこれに結合されている。

【0012】

本発明の1つの特徴において、変換システムはスカラーおよびベクトル成分の両方をグラフィック処理中に処理するように構成することができる。これを行うために、頂点データはベクトルの形態で受取られ、この後ベクトル演算が

ベクトル頂点データに関して行われる。演算論理装置および乗算論理装置または任意の別のタイプのベクトル演算モジュールがこのようなベクトル演算を行ってもよい。

【0013】

次に、ベクトル演算の出力についてスカラー演算が実行され、それによって頂点データをスカラーの形態でレンダリングしてもよい。反転論理装置または任意の別のタイプのスカラー演算モジュールがスカラー演算を行ってもよい。このようなスカラー頂点データは、その後ベクトル演算を行うためにベクトル頂点データに変換されてもよい。ベクトル演算を行うためのレジスタはまたベクトル演算の出力を記憶する。オプションとして、レジスタは、ベクトル演算の出力に基づいてベクトル頂点データを発生させるマスキング機能を備えていてもよい。

【0014】

本発明のさらに別の特徴において、グラフィックパイプラインにおけるグラフィック処理中にブレンディング、すなわち“スキニング”演算のハードウェア構成を設ける技術が使用されてもよい。パイプラインにおける処理中に、複数のマトリックスおよびそのマトリックスの1つにそれぞれ対応した複数の加重値が受取られる。処理されるべき頂点データもまた受取られる。

【0015】

その後、頂点データとマトリックスの1つとそのマトリックスに対応した加重とを乗算することによって複数の積の和が計算されることができる。その後、このような積の和は付加的な処理のために出力される。

【0016】

1実施形態において、マトリックスはモデルビューマトリックスを含んでいてもよく、付加的な処理はライティング動作を含んでいてもよい。この実施形態では、表示するために合成マトリックスもまた積の和と乗算されてもよい。さらに、そのマトリックスは逆マトリックスを含んでいてもよく、頂点データは正規ベクトルを含んでいてもよい。このような場合にもまた、ライティング動作が付加的な処理に含まれてもよい。

【0017】

【発明の実施の形態】

本発明のこれらおよび別の利点は、以下の詳細な説明を読み、その図面の種々の図を検討することによって明らかになるであろう。

上記およびその他の特徴および利点は、添付図面を参照とする以下の本発明の好ましい実施形態の詳細な説明からさらによく理解されるであろう。

図1および1Aは、従来技術を示している。図1B乃至32Cは、本発明のグラフィックパイプラインシステムを示す。

【0018】

図1Bは、本発明の1実施形態の種々のコンポーネントを示すフロー図である。示されているように、本発明は、頂点属性バッファ (VAB) 50、変換モジュール52、ライティングモジュール54、および設定モジュール57を備えたラスター化モジュール56を含む4つの主要なモジュールに分けられる。1実施形態において、上記の各モジュールは、以下においてさらに詳細に説明するように单一の半導体プラットフォーム上に配置されている。この説明において、单一の半導体プラットフォームとはただ1つの单一の半導体ベースの集積回路またはチップのことである。

【0019】

VAB50は位置、垂線、カラー、テクスチャ座標のような複数の頂点属性状態を収集し、保持するために含まれている。完成された頂点は変換モジュール52によって処理され、その後ライティングモジュール54に送られる。変換モジュール52は、照明を行うライティングモジュール54に対してベクトル発生する。ライティングモジュール54の出力は、プリミティブを設定する設定モジュールに適したスクリーンスペースデータである。その後、ラスター化モジュール56はプリミティブのラスター化を行う。変換およびライティングモジュール52および54は、コマンドが一度スタートされると常に終了されるように、コマンドレベルでのみ機能停止することを認識しなければならない。

【0020】

1実施形態において、本発明は、オープン・グラフィック・ライブラリ（商標名オープンGL）およびD3D（商標名）変換およびライティングパイプライン

ンを少なくとも部分的に使用するハードウェア構造を含んでいる。オープンGL（商標名）は2-Dおよび3-Dグラフィックイメージを既定するコンピュータ業界の標準アプリケーションプログラムインターフェース（A P I）である。このオープンGL（商標名）により、アプリケーションは任意のオペレーティングシステムにおいて任意のオープンGL（商標名）に従うグラフィックアダプタを使用して同じ効果を生成する。オープンGL（商標名）は、1組のコマンドまたは即時実行機能を特定する。各コマンドは描写アクションを指令するか、あるいは特別な効果を生じさせる。

【0021】

図2は、本発明の1実施形態によるVAB50の概略図である。示されているように、VAB50はコマンドビット200を送り、一方において頂点の属性を表すデータビット204とモードビット202とを記憶する。使用において、VAB50は頂点のデータビット204を受取り、これを出力する。

【0022】

VAB50は、複数の可能な頂点属性状態をデータビット204により受取り、記憶するように構成されている。使用において、このようなデータビット204すなわち頂点データが受取られ、VAB50中に記憶された後、頂点データはVAB50からグラフィック処理モジュール、すなわち変換モジュール52に出力される。さらに、図2Aを参照としてさらに詳細に後述する他の処理に加えて、頂点データがVAB50に入力される方法を決定するコマンドビット200がVAB50によって送られる。このようなコマンドビット200はマイクロ制御装置、CPU、データソース、またはコマンドビット200を発生できる任意の他のタイプのソースのようなコマンドビットソースから受取られる。

【0023】

さらに、モードビット202が送られ、このモードビット202は処理動作の複数のモードの状態を示す。したがって、モードビット202は、後続するグラフィック処理モジュール中で頂点データが処理される方法を決定するように適応されている。このようなモードビット202は、マイクロ制御装置、CPU、データソース、またはモードビット202を発生することのできる任意の他のタイプのソ

スのようなコマンドビットソースから受取られる。

【0024】

VAB50に関連した種々の機能が専用ハードウェア、ソフトウェアまたは任意の他のタイプの論理装置によって制御されてもよい。種々の実施形態において、モードビット202 の64、128、256または任意の他の数が使用されてもよい。

【0025】

VAB50はまた128ビットフォーマットに変換される必要のある64ビットデータのための収集地点として機能する。VAB50の入力は64ビット/サイクルであり、その出力は128ビット/サイクルである。別の実施形態において、VAB50は128ビットビットデータのための収集地点として機能してもよく、また、VAB50の入力は128ビット/サイクルまたは任意の他の組合せであってよい。さらにVAB50は複数の頂点属性のために確保されたスロットを有しており、それらは全てIEEE32ビットフロートである。このようなスロットの数はユーザの所望に応じて異なってよい。表1は、本発明によって使用される例示的な頂点属性を示している。

【0026】

表1

位置: x, y, z, w
拡散カラー: r, g, b, a
反射カラー: r, g, b
フォグ: f
テクスチャ0: s, t, r, q
テクスチャ1: s, t, r, q
垂線: n x, n y, n z
スキンウェイト: w

【0027】

動作中、VAB50は、x, yの書き込み時にz, w対を(0.0, 1.0)にデフォルトすることが可能となるため、z, wデータ対の前にx, yデータ対が書

込まれるものとして動作する。これはオープンGL（商標名）およびD3D（商標名）におけるデフォルト成分にとって重要である可能性がある。位置、テクスチャ0およびテクスチャ1のスロットは第3および第4の成分を(0, 0, 1, 0)にデフォルトすることを認識しなければならない。さらに、拡散カラースロットは第4の成分を(1, 0)にデフォルトし、テクスチャスロットは第2の成分を(0, 0)にデフォルトする。

【0028】

VAB50は、データビット204をアセンブルするために使用されるさらに別のスロット205を含んでおり、このデータビット204は変換モジュール54に送られることができ、あるいはライティングモジュール54から受取られることができ。スロット205におけるデータビット204は浮動小数点または整数フォーマットであることができる。上述したように、各頂点のデータビット204は、データビット204の処理に影響を与えるモードを表すモードビット202の関連したセットを有している。これらのモードビット202は、以下さらに詳細に説明する理由のためにデータビット204により変換およびライティングモジュール52および54を通じて送られる。

【0029】

1実施形態において、VAB50によって受取られる18の有効なVAB、変換およびライティングコマンドが存在することができる。図2Aは、本発明の1実施形態によるVAB50によって受取られることのできる種々のコマンドを示すチャートである。図2Aのチャートに示されている全てのロードおよび読み出し文脈コマンドならびにパススルーコマンドは、128ビットまでの、またはその他のサイズの1つのデータワードを転送することを認識しなければならない。

【0030】

図2Aの各コマンドは、データビット204の各セットが1つのVABアドレスの高ダブルワードまたは低ダブルワード中に書込まれるか否かを指示する制御情報を含んでいる可能性がある。さらに、ワードレベルの制御を行う2ビットの書き込みマスクが使用されてもよい。さらに、実行されるべき現在のコマンドに対するデータビット204の全てが存在していることを開始(launch)ビット

がVAB制御装置に通知してもよい。

【0031】

各コマンドは関連した機能停止フィールドを有しており、このフィールドによってルックアップは、そのコマンドが文脈メモリを読出す読出しコマンドであるか、あるいは文脈メモリに書込む書き込みコマンドであるかに関する情報を見出すことが可能になる。現在実行中のコマンドの機能停止フィールドを使用することによって、新しいコマンドは、矛盾の場合には待機させられ、あるいは進行することを可能にされることができる。

【0032】

動作において、VAB50はサイクル当たり128ビット（または他の任意のサイズ）までの1つの入力データワードを受取り、サイクル当たり128ビット（または他の任意のサイズ）までの1つのデータワードを出力することができる。ロードコマンドに対して、これは、データをVAB50中にロードして128ビットのクワド（quad）ワードを生成するのに2サイクル必要であり、それをドレインするために1サイクル必要であることを意味する。ライティングモジュール54内のスカラーメモリに対して、全クワドワードを累算する必要はなく、これらは1サイクル／アドレスでロードされることができる。1つの頂点に対して、7つのVABスロットをロードするのに14サイクルまで必要になる可能性があり、一方それらをドレインするには7サイクルあればよい。しかしながら、実行中の頂点コマンドを変更する頂点状態を更新するだけでよいことを認識すべきである。これは、ある場合には頂点位置が2サイクルで更新され、一方頂点データをドレインするのに7サイクルを要することを意味する。x, y位置の場合には、たった1サイクルあればよいことを認識しなければならない。

【0033】

図2Bは、グラフィック処理中に頂点属性をVAB50にロードし、VAB50からドレインする1つの方法を示すフローチャートである。最初に、動作210で、VAB50において頂点属性の少なくとも1つのセットが処理されるために受取られる。上述したように、頂点属性の各セットは特有で、単一の頂点に対応していることができる。

【0034】

使用において、頂点属性は動作212 でV A B 50が受取ったときにその中に記憶される。さらに、記憶された頂点属性の各セットは変換モジュール52の複数の入力バッファの対応した1つに転送される。受取られた頂点属性セットはまた、動作216 で示されているように、受取られた頂点属性が現在V A B 50中に記憶されている異なったセットの対応した頂点属性を有しているか否かを決定するため監視される。

【0035】

決定ブロック217 において記憶されている頂点属性が受取られた頂点属性に対応していると決定されたとき、動作218 に示されているように、記憶されている頂点属性は変換モジュール52の対応した入力バッファにばらばらな順序で出力される。記憶されている頂点属性が output されるとすぐに、対応した入ってきた頂点属性がV A B 50中にその場所を占有してもよい。しかしながら、対応が全く見出されない場合、動作219 に示されているように、記憶されている頂点属性の各セットは規則的な予め定められたシーケンスにしたがって変換モジュール52の対応した入力バッファに転送されてもよい。

【0036】

記憶されている頂点属性は、それが関連した開始コマンドを有している場合には、上述した方式で転送されない可能性があることを注意すべきである。さらに、上記の方法が適切に行われるためにはV A B 50の出力の帯域幅は少なくともV A B 50の入力の帯域幅でなければならない。

【0037】

図2Cは、図2Bの動作を実行するために使用される本発明のアーキテクチャを示す概略図である。示されているように、V A B 50は書き込みデータ端子W Dと、読み出しデータ端子R Dと、書き込みアドレス端子W Aと、および読み出しアドレス端子R Aとを有している。読み出しデータ端子は第1のクロック制御されたバッファ230 に結合され、データビット204 をV B A 50から出力する。

【0038】

第1のマルチプレクサ232 もまた含まれており、これはV A B 50の読み出しア

ドレス端子と第2のクロック制御されたバッファ234 とに結合された出力を有している。第1のマルチプレクサ232 の第1の入力はV A B 50の書き込みアドレス端子に結合され、一方第1のマルチプレクサ232 の第2の入力は第2のマルチプレクサ236 の出力に結合されている。論理モジュール238 は第1および第2のマルチプレクサ232 および236 と、V A B 50の書き込みアドレス端子と、第2のクロック制御されたバッファ234 の出力との間に結合されている。

【0039】

使用において、論理モジュール238 は、入ってきた頂点属性が未決定でありV A B 50においてドレインするか否かを決定する作用を行う。1実施形態において、この決定は、頂点属性が未決定であるか否かを示すビットレジスタを監視することより容易に行われることができる。入ってきた頂点属性がこの時点でV A B 50中に一致したもの有していると決定された場合、論理モジュール238 は、入ってきた頂点属性がすぐにその場所に記憶されるようにその一致した頂点属性をドレインするために第1のマルチプレクサ232 を制御する。他方、入ってきた頂点属性がこの時点でV A B 50中に一致したもの有しないと決定された場合、論理モジュール238 は、V A B 50がドレインされ、入ってきた頂点属性が、論理モジュール238 によって更新される第2のマルチプレクサ236 の入力によって順次または別のある予め定められた順序でロードされるように、第1のマルチプレクサ232 を制御する。

【0040】

その結果、V A B 50は、新しい入ってきた頂点属性がロードされる前に、多数の頂点属性をドレインする必要がない。未決定の頂点属性は、可能ならば、対応したV A B の対応したもの押し出し、それによってそれが進行することを可能にする。その結果、V A B 50は任意の順序でドレインすることができる。この能力がないと、V A B 50をドレインするのに7サイクルを要し、それをロードするのに、おそらく、さらに14サイクル要する。ロードとドレインとを重複させることにより、さらに高い性能が得られる。これは、入力バッファが空であり、V A B 50が変換モジュール52の入力バッファ中にドレインできる場合にのみ可能であることを認識しなければならない。

【0041】

図3は、本発明の1実施形態によるVAB50に関連したモードビットを示している。変換/ライティングモード情報は、モードビット202によりレジスタ中に記憶される。モードビット202は、以下において明らかになるように、変換モジュール52およびライティングモジュール54のシーケンサを駆動するために使用される。各頂点は特有であることが可能である関連したモードビット202を有しており、したがって特有に構成されたプログラムシーケンスを実行することができる。モードビット202は一般にグラフィックAPIに直接マップすることができるが、それらのあるものは導出されてもよい。

【0042】

1実施形態において、図3のアクティブな光ビット(LIS)は隣接してもよい。さらに、パススルービット(VPAS)は、オンにされたときに頂点データがスケールおよびバイアスと共に通過され、変換もライティングも行われないという点で特有である。VPASが真(true)である場合に使用される可能なモードビット202はテクスチャ分割ビット(TDV0, 1)およびフォグビット(商標名D3Dにおいてフォグ値を抽出するために使用される)である。したがって、VPASは予め変換されたデータに対して使用され、TDV0, 1は商標名D3Dの文脈では円筒状ラップモードを処理するために使用される。

【0043】

図4は、本発明の1実施形態の変換モジュールを示している。示されているように、変換モジュール52は6つの入力バッファ400によってVAB50に接続されている。1実施形態において、各入力バッファ400はサイズが7*128ビットである。6つの入力バッファ400はそれぞれ7つのクワドワードを記憶することができる。そのような各入力バッファ400は、パスデータが位置データと重複されていることを除いて、そのレイアウトがVAB50と同じである。

【0044】

1実施形態において、入力バッファ400がロードされた前の段階からデータが変化しているか否かを示すように各入力バッファ400の各属性に対して1つのビットが指定されていてもよい。この設計によって、各入力バッファ400は変化

したデータに関してのみロードされてもよい。

【0045】

変換モジュール52はさらに、ライティングモジュール54中の6つの出力頂点バッファ402に接続されている。出力バッファは第1のバッファ404と、第2のバッファ406と、および第3のバッファ408とを含んでいる。以下から明らかになるように、第3のバッファ408の内容、すなわち位置、テクスチャ座標データ等はライティングモジュール54では使用されない。しかしながら、第1のバッファ404および第2のバッファ406の両者は光線およびカラーデータをライティングモジュール54に入力するために使用される。ライティングモジュールは2つの読み出し入力を処理するように構成されているため、2つのバッファが使用される。データは読み出しコンフリクト等に関する問題を回避するように構成されていることを認識すべきである。

【0046】

さらに、変換モジュール52には文脈メモリ410およびマイクロコードROMメモリ412が結合されている。変換モジュール52はオブジェクト空間頂点データをスクリーン空間に変換して、ライティングモジュール54が必要とするベクトルを生じさせる作用をする。変換モジュール52はまたスキニング(skinning)およびテクスチャ座標を処理する。1実施形態において、変換モジュール52は平行して4つのフロートを処理する128ビット設計であってもよく、4項ドット積を行うために最適化されていてもよい。

【0047】

図4Aは、変換モジュール52中で多数のスレッドを実行する本発明の1実施形態による方法を示すフローチャートである。動作において、変換モジュール52はインターリーブすることにより3つの頂点を平行して処理することができる。したがって、書き込みおよびそれに後いて文脈メモリ410からの読み出しのようなコマンドの間に機能停止状況が生じなければ、3つのコマンドが同時に平行して実行することができる。3つの実行スレッドは互いに独立しており、任意コマンドであることができる。これは、全ての頂点が特有の対応したモードビット202を含んでいるためである。

【0048】

図4Aに示されているように、多数のスレッドを実行する方法は、動作420において実行されるべき現在のスレッドを決定することを含んでいる。この決定は、グラフィック処理モジュールがある動作の終了に必要とするサイクルの数を識別して、そのサイクルを追跡することにより行われることができる。サイクルを追跡することによって、各スレッドはあるサイクルに割当てられ、それによって現在のサイクルに基づいて現在のスレッドを決定することができる。しかしながら、このような決定は、効果的であると思われる任意の所望の方式で行われてもよいことを認識しなければならない。

【0049】

次に動作422において、現在のサイクル中に実行されるべきスレッドに関連した命令が、対応したプログラムカウンタ番号を使用して検索される。その後、この命令は動作424においてグラフィック処理モジュールに関して実行される。

【0050】

1 使用例において、この方法は、最初に、第1のプログラムカウンタによって第1の命令またはコードセグメントにアクセスすることを含む。上述のように、このようなプログラムカウンタは第1の実行スレッドと関連している。次に、第1のコードセグメントがグラフィック処理モジュールにおいて実行される。すぐに明らかになるように、このようなグラフィック処理モジュールは加算器、乗算器または任意の他の機能装置あるいはその組合せの形態をとることができる。

【0051】

グラフィック処理モジュールは実行を完了するために2以上のクロックサイクルを必要とするため、第1のコードセグメントの実行後1クロックサイクル経過してすぐに第2のコードセグメントが第2のプログラムカウンタによってアクセスされてもよい。第2のプログラムカウンタは第2の実行スレッドと関連しており、各実行スレッドが特有の頂点を処理する。

【0052】

その後、第2のコードセグメントは、グラフィック処理モジュール中での第1のコードセグメントの実行の終了前に、グラフィック処理モジュールにおいて

実行を開始してもよい。使用において、グラフィック処理モジュールは出力を発生するために全てのスレッドのそれぞれに対して予め定められた数のサイクルを必要とする。したがって、全ての予め定められた数のサイクルのそれぞれに対してこの例の種々のステップが反復されてもよい。

【0053】

この技術は、従来技術より優れた多くの利点を提供する。もちろん、本発明の機能装置はさらに効率的に使用される。さらに、多数スレッド方式が使用されると仮定された場合、統御コードがより効率的に書込まれることができる。

【0054】

たとえば、答えを出力するのに3クロックサイクルを必要とする乗算器がグラフィック処理モジュールに含まれている場合、 $a = b * c$ および $d = e * a$ のような後続する動作の間に2つの無動作コマンドを含むことが必要となる。その理由は、3クロックサイクル後まで “a” が利用できないためである。しかしながら、この実施形態では、コードが $a = b * c$ の直後に $d = e * a$ を呼出すだけでもよい。それは、このようなコードは3クロックサイクルごとに1回呼出される3つの実行スレッドの1つとして実行されると仮定されることができるからである。

【0055】

図4Bは、図4Aの方法が行われる手順を示すフロー図である。示されているように、各実行スレッドは、命令メモリ452において命令またはコードセグメントにアクセスするために使用される関連したプログラムカウンタ450を有している。その後、このような命令は加算器456、乗算器454および、または反転論理装置またはレジスタ459のようなグラフィック処理モジュールを動作するために使用されてもよい。

【0056】

上記の処理モジュールの2以上のものがタンデム方式で使用される状況に適応させるために、グラフィック処理モジュール間において1以上のコードセグメント遅延素子457が使用される。3スレッドフレームワークが使用される場合、3クロックサイクルコードセグメント遅延素子457が使用される。1実施形態に

おいて、加算命令が乗算命令に後続した場合に、コードセグメント遅延素子457が使用される。このような場合、乗算器456が出力を発生するために十分な時間が確実に経過するように、乗算命令の実行後3クロックサイクル経過するまで加算命令は実行されない。

【0057】

各命令の実行後、現在の実行スレッドのプログラムカウンタ450が更新され、次の実行スレッドのプログラムカウンタが関連した命令にアクセスするためにラウンドロビンシーケンスでモジュール458により呼出される。プログラムカウンタは、インクリメント、ジャンプ、呼出および復帰、テーブルジャンプ、および、またはディスパッチを含む任意の方式で使用されることができるが、それに限定されないことを認識しなければならない。ディスパッチとは、受取られたパラメータに基づいてコードセグメント実行の開始地点を決定することである。さらに、この多数スレッド実行フレームワークに関連した原理は、本発明のグラフィック処理パイプラインのライティングモジュール54にも適用可能であることを認識することが重要である。

【0058】

3スレッドフレームワークが使用される場合、任意のある時間に各スレッドが1つの入力バッファおよび1つの出力バッファに割当てられる。これによって、3つのコマンドを処理しながら、さらに3つのコマンドをデータと共にロードすることが可能になる。入力バッファおよび出力バッファは、以下において図27および28を参照として説明する方式によりラウンドロビンシーケンスで割当てられる。

【0059】

したがって、実行スレッドは時間的および機能的にインターリープされる。これは、各機能装置が3つのステージにパイプラインされ、各スレッドがいつでも1つのステージを占有していることを意味する。1実施形態において、3つのスレッドは常に同じシーケンスで実行するよう、すなわち0、1、3に設定されてもよい。概念上、スレッドは $t = \text{クロックモジュロ } 3$ において機能装置に入力される。機能装置が動作し始めると、結果を出力するのに3サイクルを要し（

6サイクルを必要とする I L Uを除いて)、その時同じスレッドは再びアクティブである。

【0060】

図5は、本発明の1実施形態による図4の変換モジュール52の機能装置を示している。示されているように、頂点データを受取るために V A B 50に結合するように構成された入力バッファ400 が含まれている。

【0061】

メモリ論理装置 (M L U) 500 は、入力バッファ400 の出力に結合された第1の入力を有している。オプションとして、M L U500 の出力は、その第1の入力に結合されたフィードバックループ502 を有していてもよい。

【0062】

演算論理装置 (A L U) 504 もまた設けられており、このA L Uの第1の入力はM L U500 の出力に結合されている。A L U504 の出力はさらに、その第2の入力に接続されたフィードバックループ506 を有している。このようなフィードバックループ502 はさらに、それに結合された遅延素子508 を有していてもよい。A L U504 の出力には、レジスタ装置510 の入力が結合されている。レジスタ装置510 の出力は、M L U500 の第1および第2の入力に結合されていることを認識しなければならない。

【0063】

反転論理装置 (I L U) 512 が設けられており、このI L U512 は、逆数または逆平方根演算を行うためにA L U504 の出力に結合された入力を含んでいる。別の実施形態において、I L U512 はレジスタ装置510 の出力に結合された入力を含んでいてもよい。

【0064】

さらに、変換またはスマアリング (s m e a r i n g) モジュール514 が含まれており、このモジュール514 は I L U512 の出力とM L U500 の第2の入力との間に結合されている。使用において、この変換モジュール514 はスカラーポイントデータをベクトルポイントデータに変換するように機能する。これはスカラーデータをベクトルと乗算して、乗算器および、または加算器が処理するベクトル演算

子にすることによって行なわれる。たとえば、スカラーAは、変換後、ベクトル(A, A, A, A)になってもよい。別の実施形態では、スマアリングモジュール514はMLU500と関連したマルチプレクサまたは本発明の任意の他のコンポーネント中に含まれていてもよい。オプションとして、レジスタ516はILU512の出力と変換装置514の入力との間に結合されていてもよい。さらに、このようなレジスタ516はスレッド(thread)されてもよい。

【0065】

メモリ410は、MLU500の第2の入力とALU504の出力とに結合されている。とくに、メモリ410はMLU500の第2の入力に結合された読出し端子を有している。さらに、メモリ410はALU504の出力に結合された書き込み端子を有している。

【0066】

メモリ410は、頂点データを処理するために入力バッファ400、MLU500、ALU504、レジスタ装置510、ILU512および変換モジュール514と共に使用されるために複数の定数および変数が記憶されている。このような処理には、オブジェクト空間頂点データをスクリーン空間頂点データに変換し、ベクトルを発生すること等が含まれる。

【0067】

最後に、出力コンバータ518はALU504の出力に結合されている。出力コンバータ518は、処理された頂点データがこれに出力されるように出力バッファを介してライティングモジュール54に結合されている。ILUを除く全てのデータ通路は128ビット幅であるように設計されてもよく、あるいは別のデータ通路幅が使用されてもよい。

【0068】

図6は、本発明の1実施形態による図5の変換モジュール52のMLU500の概略図である。示されているように、変換モジュール52のMLU500は、並列に結合された4つの乗算器600を含んでいる。

【0069】

変換モジュール52のMLU500は、3つの異なった方式で2つの4成分ベク

トルを乗算するか、あるいは1つの4成分ベクトルをパス (pass) することができる。MLU500 は、多重演算を行うことができる。表2は、変換モジュール52のMLU500 に関連したこのような演算を示している。

【0070】

表2

【数1】

```

CMLU_MULT  o[0] = a[0]*b[0],o[1] = a[1]*b[1],o[2] = a[2]*b[2],o[3] = a[3]*b[3]
CMLU_MULA  o[0] = a[0]*b[0],o[1] = a[1]*b[1],o[2] = a[2]*b[2],o[3] = a[3]
CMLU_MULB  o[0] = a[0]*b[0],o[1] = a[1]*b[1],o[2] = a[2]*b[2],o[3] = b[3]
CMLU_PASA  o[0] = a[0],o[1] = a[1],o[2] = a[2],o[3] = a[3]
CMLU_PASB  o[0] = b[0],o[1] = b[1],o[2] = b[2],o[3] = b[3]

```

【0071】

表3には、可能なAおよびB入力が示されている。

【0072】

表3

MA_M	MLU
MA_V	入力バッファ
MA_R	RLU (MB_Rと共有された)
MB_I	ILU
MB_C	文脈メモリ
MB_R	RLU (MA_Rと共有された)

【0073】

表4は、クロス乗積に対して使用されることのできる回転オプションを示している。

【0074】

表4

MR_NONE	変更なし
MR_ALBR	A[XYZ]ベクトルを左に、B[XYZ]ベクトルを右に回転する
MR_ARBL	A[XYZ]ベクトルを右に、B[XYZ]ベクトルを左に回転する

【0075】

図7は、本発明の1実施形態による図5の変換モジュール52のA L U504の概略図である。示されているように、変換モジュール52のA L U504は、並列／直列に結合された3つの加算器700を含んでいる。使用において、変換モジュール52のA L U504は2つの3成分ベクトルを加算し、1つの4成分ベクトルをパスし、あるいはベクトル成分を出力を横切ってスマア(s m e a r)することができる。表5は、変換モジュール52のA L U504が行うことのできる種々の演算を示している。

【0076】

表5

【数2】

CALU_ADDA	$o[0] = a[0] + b[0], o[1] = a[1] + b[1], o[2] = a[2] + b[2], o[3] = a[3]$
CALU_ADDB	$o[0] = a[0] + b[0], o[1] = a[1] + b[1], o[2] = a[2] + b[2], o[3] = b[3]$
CALU_SUM3B	$o[0123] = b[0] + b[1] + b[2]$
CALU_SUM4B	$o[0123] = b[0] + b[1] + b[2] + b[3]$
CALU_SMRB0	$o[0123] = b[0]$
CALU_SMRB1	$o[0123] = b[1]$
CALU_SMRB2	$o[0123] = b[2]$
CALU_SMRB3	$o[0123] = b[3]$
CALU_PASA	$o[0] = a[0], o[1] = a[1], o[2] = a[2], o[3] = a[3]$
CALU_PASB	$o[0] = b[0], o[1] = b[1], o[2] = b[2], o[3] = b[3]$

【0077】

表6は、変換モジュール52のA L U504のAおよびB入力を示している。

【0078】

表6

A A _ A	A L U (1つの命令遅延)
A A _ C	文脈メモリ
A B _ M	M L U

無変更、Bの否定、Aの否定を行なうことによりAおよびB入力の符号ビットを修正することもまた可能であり、ここでA、Bは絶対値である。A L U504がスカラー頂点データを出力した場合、このスカラー頂点データは、各出力がスカラー頂点データを表しているという意味で出力を横切ってスマアされていることを認識しなければならない。M L U500およびA L U504のバス制御信号のそれ

それが演算中全ての特殊値処理をディスエーブルすることができる。

【0079】

図8は、本発明の1実施形態による図5の変換モジュール52のベクトルレジスタファイル510の概略図である。示されているように、ベクトルレジスタファイル510は4組のレジスタ800を含んでおり、各レジスタ800は対応したマルチプレクサ802の第1の入力に結合された出力と、対応したマルチプレクサ802の第2の入力に結合された入力を有している。

【0080】

本発明の1実施形態において、ベクトルレジスタファイル510はスレッドされている。すなわち、ベクトルレジスタファイル510の3つのコピーが存在し、各スレッドがそれ自身のコピーを有している。1実施形態では、各コピーは8つのレジスタを含んでおり、その各レジスタはサイズが128ビットであり、4つのフロートを記憶することができる。ベクトルレジスタファイル510はALU504から書込まれ、その出力はMLU500にフィードバックされる。ベクトルレジスタファイル510はサイクル当たり1回の書き込みおよび1回の読み出しを行なう。

【0081】

動作において、各レジスタコンポーネントへの書き込み動作を個々にマスクすることができる。ベクトルレジスタファイル510は、書き込みアドレスが読み出しアドレスと同じである場合、入力から出力へのバイパス路511によってゼロレイテンシーを示す。この場合、マスクされていないコンポーネントはレジスタから取出され、マスクされたコンポーネントはバイパスされる。このように、ベクトルレジスタファイル510はコンポーネント単位でベクトルを生成し、あるいはALU SMR演算(表5参照)と共にベクトル成分の順序を変更することに対して非常に有用である。一時的な結果はまたベクトルレジスタファイル510中に記憶されることができる。

【0082】

図9は、本発明の1実施形態による図5の変換モジュール52のILU512の概略図である。示されているように、変換モジュール52のILU512は浮動小数点の逆数($1/D$)および逆平方根($1/D^{1/2}$)を発生することができる。

きる。このような演算を行なうために、2つの反復処理のいずれか一方が小数部に関して実行されてもよい。このような処理は任意の所望の専用ハードウェアにより実行されてもよく、以下に示されている：

逆数 ($1/D$)	逆平方根 ($1/D^{1/2}$)
$x_n + 1 = x_n (2 - x_n * D)$	$x_{n+1} = (1/2) * x_n (3 - x_n^2 * D)$
(1) x_n (速度) に対する表検索	x_n (速度) に対する表検索
x_n	$x_n * x_n$
(2) 第1回目の反復：乗算一加算	第1回目の反復：乗算一加算
$2 - x_n * D$	$3 - x_n^2 * D$
(3) 第1回目の反復：乗算	第1回目の反復：乗算
$x_n (2 - x_n * D)$	$(1/2) * x_n (3 - x_n^2 * D)$
(4) 第2回目の反復：演算なし	第2回目の反復：2乗
$x_n + 1$ をパス	x_{n+1}^2
(5) 第2回目の反復：乗算一加算	第2回目の反復：乗算一加算
$2 - x_{n+1} * D$	$3 - x_{n+1}^2 * D$
(6) 第2回目の反復：乗算	第2回目の反復：乗算
$x_{n+1} (2 - x_{n+1} * D)$	$(1/2) * x_{n+1} (3 - x_{n+1}^2 * D)$

示されているように、2つの処理は類似しており、簡単な設計を行なっても差しつかえない。この反復は、しきい値精度が満足されるまで繰り返されることを認識しなければならない。

【0083】

動作において、ILU512 は逆数演算および逆平方根演算を含む2つの基本的な演算を行なう。他の装置とは異なり、それは出力を発生するために6サイクルを必要とする。その入力はスカラーであり、したがって出力もそうである。前述したように、ILU512 の出力におけるスレッド保持レジスタ516 は、有効な結果が発生される次の回まで結果をラッチするように当てにされている。さらに、スカラー出力は、MLU500 に供給される前にベクトルにスマアされる。反転装置512 は、約22小数部ビット範囲内までの正確な IEEE (米国電気電子技術者協会) 出力を発生するために検索表および2つのパスNewton-Rap

h s o n 反復を使用する。表7は、変換モジュール52のI L U512 によって行なわれることのできる種々の演算を示している。

【0084】

表7

C I L U_I N V	$o = 1. 0 / a$
C I L U_I S Q	$o = 1. 0 / \sqrt{a}$
C I L U_C I N V	$o = 1. 0 / a$ (レンジクランプにより)
C I L U_N O P	出力なし

表7の上述したレンジクランプ反転演算は、クリッピング演算がラスター化モジュール56により処理されることを可能にするために使用されてもよい。座標はスクリーン空間に直接変換され、これは、均質のクリップスペースがほぼ0. 0である場合に問題を結果的に生じさせる可能性が高い。各除算において $1. 0 / 0. 0$ による乗算を回避するために、 $1 / w$ 計算が最小および最大ベキ指数にクランプされる。

【0085】

使用において、図5に示されている文脈メモリ410 は、クワドワードだけを使用して読み出しおよび書き込みを行なう。このメモリはM L U500 またはA L U504 によって各サイクルごとに読み出されることができ、A L U504 によって書き込まれることができる。メモリ読み出しあはサイクル当たり1度だけ可能である。読み出しが必要である場合には、それは命令の開始時に行なわれ、それから3サイクル後にA L U504 にパイプラインされる。文脈メモリ410 は必ずしもスレッドされなくてよい。

【0086】

図10は、本発明の1実施形態による図5の変換モジュール52の出力コンバータ518 の出力アドレスのチャートである。出力コンバータ518 は出力を適切な目的に導き、データのビット精度を変更し、性能を増加させるためにあるデータ攪拌 (swizzling) を行なうことができる。ライティングモジュール54に送られる予定である全てのデータは、S I E 8 M 1 3として編成された22ビット浮動小数点フォーマット (1符号、8ベキ指数、13小数部ビット) に丸

められる。ライティングモジュール54における図4に示されているような目的地バッファ402はスレッドされる。

【0087】

データ攪拌は、ベクトルを発生しているときに有用である。このような技術により、ベクトルを生成する場合に損失を生じずに距離ベクトル（ $1, d, d * d$ ）を発生することが可能となる。距離ベクトルはフォグ、地点パラメータおよび照明減衰に対して使用される。これは、アイベクトルおよび照明方向ベクトルにより行なわれる。表8は、このようなベクトルに関連した種々の演算を示している。以下の表において、ベクトルを2乗するとは $d^2 = d \cdot t [(x, y, z), (x, y, z)]$ である d^2 を (x, y, z) のwコンポーネント中に記憶することを指していることを認識しなければならない。

【0088】

表8

- (1) ベクトルを2乗する $(x, y, z, d * d)$ ($d * d$ をV B U Fに出力し、 $1, 0$ をV B U Fに出力する)
- (2) $d * d$ の逆平方根を発生する $(1/d)$
- (3) ベクトルを正規化する $(x/d, y/d, z/d, d)$ ($x/d, y/d, z/d$ をW B U Fに出力し、 d をV B U Fに出力する)

本発明において行なわれた数学的計算は常に I E E E 方式に従つたものである必要はないことを認識しなければならない。たとえば、任意の数により乗算された“0”は“0”をレンダリングすると仮定されることがある。これは、 $d = 0$ である $d = d^2 * 1 / (d^2)^{1/2}$ のような式を処理する場合にとくに有用である。上記の仮定を行わないと、このような式はエラーを生じ、したがって関連した計算を行なうときに問題が発生する。

【0089】

図11は、本発明の1実施形態による図5の変換モジュール52のマイクロコード編成を示す図である。変換モジュールのマイクロコードは、44ビットの総帯域幅を形成する15のフィールドに構成されてもよい。フィールドは、装置のデータフローを一致させるために遅延されてもよい。M L U 500 の演算はゼロの

遅延で実行される。ALU演算は1の遅延で実行され、RLUの出力演算は2の遅延で実行される。各遅延は3サイクルと等価である。

【0090】

図12は、本発明の1実施形態による図5の変換モジュール52のシーケンサ1200の概略図である。図12に示されているように、変換モジュール52のシーケンサ1200は、処理動作の複数のモードの状態を示すモードビットをVAB50から受取るように構成されたバッファ1202を含んでいる。

【0091】

メモリ412もまた含まれており、このメモリ412は、モードの状態にしたがって処理動作を行なうようにそれぞれ構成されたコードセグメントを記憶することができる。シーケンシングモジュール1206はメモリ412と制御ベクトルモジュール1205との間に結合されており、この制御ベクトルモジュール1205はバッファ1202に結合され、モードビット202から得られた制御ベクトルに基づいてメモリ412中の複数のアドレスを識別する。シーケンシングモジュール1206はさらに、データを出力バッファ1207に転送するように変換モジュール52を動作するために使用されることのできるコードセグメントを検索するためにメモリ412中のアドレスにアクセスするように構成されている。

【0092】

図13は、図12の変換モジュール52のシーケンサ1200の使用に関連した種々の動作を詳細に示すフローチャートである。示されているように、シーケンサ1200は、変換またはライティング動作におけるグラフィック処理をシーケンス化するように構成されている。動作1320において、処理動作の複数のモードの状態を示すモードビット202が最初に受取られる。1実施形態において、モードビット202はソフトウェア駆動装置から受取られてもよい。

【0093】

その後、動作1322において、メモリ中の複数のアドレスがモードビット202に基づいて識別される。その後、動作1324において、そのモードの状態にしたがって処理動作を行なうようにそれぞれ構成されたコードセグメントを検索するために、メモリ中のこのようなアドレスがアクセスされる。続いて、動作1326に示

されているように、頂点データを処理するために変換またはライティングモジュールによりコードセグメントが実行される。

【0094】

図14は、図12の変換モジュール52のシーケンサ1200のシーケンシングモジュール1206の動作を詳細に示すフロー図である。示されているように、複数のモードレジスタ1430はそれぞれ、単一の頂点に対応するモードビット202の特有のセットを含んでいる。モードレジスタ1430は、図4Aおよび4Bを参照として上述した方式での多数の実行スレッドの実行を可能にするためにラウンドロビンシーケンスでポールされることを認識すべきである。

【0095】

現在の実行スレッドが選択されると、モードビット202の対応したグループは動作1432でデコードされる。動作1432においてモードビット202がデコードされると、対応した頂点データを処理する特定のコードセグメントがROM1404においてアクセスされたか否かをそれぞれ示す複数のビットを含む制御ベクトルが供給される。

【0096】

コードセグメントがROM1404でアクセスされ実行されるべきであるか否かを決定するとき、ポインタ動作1436は現在のスレッドポインタをインクリメントして、次の実行スレッドを開始し、それによって類似の動作を継続するよう第2のグループモードビット202を獲得する。これはラウンドロビンシーケンスで各スレッドに対して継続される。

【0097】

制御ベクトルが一度、モードビット202の特定のグループに対して形成されると、優先度エンコーダ動作1438は次の“1”またはエネーブルされた制御ベクトルのビットを決定し、識別する。このようなビットが発見されると、優先度エンコーダ動作1438は実行のために、制御ベクトルのエネーブルビットに対応するアドレスをROM1404中に生成する。

【0098】

残りのスレッドを処理した後、およびモードビットがデコードされ、制御ベ

クトルが再度有効になった後、モードビット202 の最初のグループに戻るとき、マスキング動作1434は先の“1”または前に識別されたエネーブルされたビットをマスクするために使用される。これはマスク動作1434後に全ての残りのビットの解析を可能にする。

【0099】

前述のプロセスは以下の表を使用して示されている。表9はサブジェクト頂点データについて実行される複数の式を示している。

表9

【数3】

$$\begin{aligned}
 R &= (a) \\
 R &= (a + d^*e) \\
 R &= (a + b^*c + f) \\
 R &= (a + b^*c + d^*e) \\
 R &= 1.0/(a) \\
 R &= 1.0/(a + d^*e) \\
 R &= 1.0/(a + b^*c + f) \\
 R &= 1.0/(a + b^*c + d^*e)
 \end{aligned}$$

【0100】

示されているように、反転演算に加えて加算される積には4つの可能性が存在する (a, b * c, d * e, f および 1 / x)。次に、モードフィールドが規定される。表10はモードフィールドの対、mode.y と mode.z を示し、それぞれ表9の演算の予め定められたセットに割当てられている。

【0101】

表10

【数4】

mode.y[4] 0: R = a
 1: R = a + d^*e
 2: R = a + b^*c + f
 3: R = a + b^*c + d^*e

mode.z[2] 0: R = R
 1: R = 1.0/R

【0102】

その後、各演算は関連するアドレスと共にメモリに位置付けされる。表1-1は関連する演算をそれぞれ有する複数のメモリアドレスを示している。また制御ベクトル定義のセットも示されている。

【0103】

表1-1

【数5】

```

ROM[0]: R = a
ROM[1]: R = R + b*c
ROM[2]: R = R + d*e
ROM[3]: R = R + f
ROM[4]: R = 1.0/R

cv[0] = 1;
cv[1] = (mode.y==2 || mode.y==3) ? 1 : 0;
cv[2] = (mode.y==1 || mode.y==3) ? 1 : 0;
cv[3] = (mode.y==2) ? 1 : 0;
cv[4] = (mode.z==1) ? 1 : 0;

```

【0104】

表1-2は1例の実行を示している。

【0105】

表1-2

$R = a + d * e$ は以下に対応する：

mode. y = 1 ;

mode. z = 0 ;

これは以下の制御ベクトルを与える：

cv [0] = 1 ;

cv [1] = 0 ;

cv [2] = 1 ;

cv [3] = 0 ;

cv [4] = 0 ;

実行

第1のサイクル：

cv [0] は TRUE であるので、ROM [0] を実行
 制御ベクトルにさらに多くの TRUE 値が存在するので、プログラムを終了
 しない

第2のサイクル：

cv [1] は FALSE であるので、観察し続ける
 cv [2] は TRUE であるので、ROM [2] を実行
 制御ベクトルには TRUE 値がもはや存在しないので、プログラムを終了
 する。

【0106】

このようにして、変換モジュール52のシーケンサ1200はスレッドされたモードビット202 から得られるスレッドされた制御ベクトルをステップし、対応する制御ベクトルビットが “TRUE” に設定されるあらゆる ROM アドレスを実行する。制御ベクトルは ROM と同一の長さを有する。シーケンサ1200は 1 つの “1” のレート、または予め定められたサイクル数毎にエネーブルされたビットで任意の制御ベクトルをステップできる。モードビット202 を使用しないコマンドはその簡潔性のためにオンザフライマイクロコードにより実行される。

【0107】

このような状態をモードビット202 の特有のストリングにより表示することによって、種々の動作の状態を決定するためにグラフィック処理ハードウェアの複数のイフーゼン (if - then) 節を実行することは必要ではない。改良された性能はそれによって与えられる。概念的に、これはプログラム言語のイフ節がシーケンサ1200へ移動するかのようであり、シーケンサ1200はモードビット202 により示されるように “FALSE” 状態で即時に命令をスキップする。

【0108】

前述したように、コードセグメントは ROM に記憶され、これはモードビットにより識別される動作の種々の状態を処理することができる。1 実施形態では、別々のコードセグメントはモードビットにより示される各動作を処理するために検索される。その代りとして、1 つの包括性コードセグメントは可能であるそれぞれまたは幾つかの動作の組合せを処理するために書込まれてもよい。しか

しながら、各動作の組合せでこのような大きいコードセグメントを生成することは付加的なコードスペースを必要とし、それ故、普通に使用される動作の組合せだけでコードセグメントをモジュール化することが有効であることに注意する。

【0109】

モードビット202は一度頂点が実行を開始すると変化しないので、制御ベクトルの生成はシーケンサに入る前に1つの頂点毎に1度実行されさえすればよい。しかしながら、これについての例外が動作が反復されるライティングのような幾つかのケースで生じる。最後の頂点命令が発見されるとき、シーケンス信号の終了（EOS）が表明される。これは入力および出力バッファの状態を変更し、図28Aと28Bを参照して説明した方法で次のコマンドの開始を可能にするために使用される。EOS信号は命令が処理される方法と類似の目的地バッファを解除するために遅延されるパイプラインであることに注意する。図4Bを参照する。

【0110】

図14Aはグラフィック処理中のスカラーおよびベクトル頂点データの管理を一体化するために使用される本発明の種々の機能コンポーネントを示したフロー図である。示されているように、1つの機能アスペクト1440はベクトル頂点データを処理モジュール、即ち加算器、乗算器等へ入力し、ベクトル頂点データを出力することを含んでいる。別の機能アスペクト1442では、ベクトル頂点データはベクトル処理モジュール、即ち加算器、乗算器等により処理され、これは再度ベクトル頂点データへ変換されるかスマアされるスカラー頂点データを出力する。

【0111】

さらに別の機能アスペクト1444では、ベクトル頂点データはマスクされ、それによってスカラー頂点データに変換され、その後、これはベクトル頂点データを生成する目的で、メモリ、即ちレジスタ論理装置中に記憶される。さらに別の機能アスペクト1446では、スカラー頂点データはベクトル処理モジュール、即ち加算器、乗算器等により抽出され、これはスカラー処理モジュール、即ち反転論

理装置により処理され、スカラー頂点データをレンダリングする。このスカラー頂点データは再度ベクトル頂点データに変換される。

【0112】

図14Bは図5の変換モジュール52に対応している図14Aに示されている本発明の機能コンポーネントの1つの可能な組合せ1451を示すフロー図である。機能アスペクト1444および1446は図4Bを参照して前述した方法と類似の方法で関連する遅延を有することに注意すべきである。図14Cは図14Aに示されている本発明の機能コンポーネントの別の可能な組合せ1453を示すフロー図である。

【0113】

マルチプレクサは図14A-14Cの機能モジュール中のベクトル頂点データからスカラー頂点データを抽出する。このようなマルチプレクサは種々の機能モジュールによる処理前に必要とされる任意のデータのスウィズリングに対しても応答可能である。1実施形態では、マルチプレクサはベクトルの頂点データを通過し回転することができ、他の処理用のALU等の他のグラフィック処理モジュールに依存する。さらに別の実施形態では、マルチプレクサはペナルティなしで独立して属性を任意選択的に再配置することができる。

【0114】

図14Dは特定用途向け集積回路(ASIC)のようなハードウェア構造によりグラフィックパイプラインにおけるグラフィック処理中に変換システムがブレンディングまたはスキン動作を行うように構成されている方法を示している。パイプラインでの処理中に、動作1470では、複数のマトリックス、それぞれ1つのマトリックスに対応する複数の加重値および頂点データが受信される。付加的なマトリックスのセットは正規の頂点データで必要とされる可能性があることに注意すべきである。

【0115】

続いて、動作1472では、複数の積の和がその後計算され、各積は頂点データと、1つのマトリックスと、そのマトリックスに対応する加重との乗算により計算される。このような積の和はその後、さらに処理を行うために動作1474で出力

される。

【0116】

要約すると、以下の積の和が計算される。

式# 1

$$i = 1 \cdots x \text{ に対して } v' = \sum w_i * M_i * v$$

ここで v = 入力された頂点データ、

w = 加重値、

M = マトリックス、

x = マトリックスの数、

v' = 処理されるモジュールへ出力される頂点データ

式# 2

$$i = 1 \cdots x \text{ に対して } n' = \sum w_i * I_i * n$$

ここで n = 入力された頂点データ（正規ベクトル）、

w = 加重値、

I = 反転マトリックス（逆転置マトリックス）、

x = 反転マトリックスの数、

n' = 処理モジュールへ出力される頂点データ（正規ベクトル）

式# 3

$$V_s = [O_x, O_y, O_z, \phi] ' + 1 / (v''_w) * [(v''_x), (v''_y), (v''_z), 1] '$$

ここで $v'' = C * v'$ 、

v' = 式# 1 からの積の和、

$$C = [S_x, S_y, S_z, 1] ' * P$$

P = 投影マトリックス、

v_s = 表示目的のスクリーンベクトル、

O = ビューポートオフセット、

S = ビューポートスケール

【0117】

前述した加重 w_i を表す方法が多数存在することに注意すべきである。例えば

式#1と#2では、 $i = 1 \dots (x-1)$ では w_x (w_i 、ここでは $i = x$) は式 $1 - \sum w_i$ により計算されることが言われている。このようにして加重 w_i を表すことにより、全ての加重 w が 1 に合計されることが確実にされる。

【0118】

1 実施形態では、マトリックスはモデルビューマトリックス (M) を含み、積の和 (v') はライティング動作によりさらに処理されるために出力される (式1参照)。この積の和 (v') はまた合成マトリックス (C) の使用によって表示目的で別の積の和 (v_s) を生成するためにも使用される (式3参照)。マトリックスは反転マトリックス (I) を含み、頂点データは正規ベクトルデータ (n) を含む。このようなケースでは、附加的な処理はライティング動作を含む (式#2参照)。

【0119】

図15は本発明の1実施形態によるライティングモジュール54の概略図である。示されているように、ライティングモジュール54は変換モジュール52が頂点データを出力するバッファ402を含んでいる。示されているように、バッファ408は通路1501によりライティングモジュール54をバイパスする。さらにライティングモジュール54には文脈メモリ1500とマイクロコードROMメモリ1502に結合されている。

【0120】

ライティングモジュール54はフォッグおよびポイントパラメータに加えてライティングを処理するように構成されている。使用において、ライティングモジュール54はバッファバイパス経路1501を制御し、拡散、ポイントサイズ、スペキュラー出力色およびフォッグ値を計算する。ライティングモジュール54は変換モジュール52と同一のモードビット202を使用することに注意すべきである。

【0121】

ライティングモジュール54はさらに変換モジュール52に関してそれ程正確性を必要とせず、それ故、3ワードで組織される22ビット浮動小数点値 (1.8.13フォーマット) を処理する。第3のバッファ408のデータは128ビットであるので、これはライティングモジュール54周辺のバイパス経路1501を使用する。ラ

イティングモジュール54は事象駆動され、同時に図4Aと4Bを参照して前述した変換モジュール52と類似の方法で3つのスレッドを実行する。ライティングモジュール54は外部ソースからコマンド発信許可を必要とすることに注意しなければならない。

【0122】

図16は本発明の1実施形態による図15のライティングモジュール54の機能装置を示す概略図である。示されているように、変換システムに結合されてそこから頂点データを受信するように構成されている入力バッファ402が含まれている。前述したように、入力バッファ402は第1の入力バッファ404、第2の入力406、第3の入力バッファ408を含んでいる。第1のバッファの入力404、第2の入力バッファ406、第3の入力バッファ408の入力は変換モジュール52の出力に結合されている。バイパスの目的で、第3の入力バッファ408の出力は遅延素子1608によりライティングモジュール54の出力に結合されている。

【0123】

さらに、第1の入力バッファ404の出力に結合されている第1の入力と、第2の入力バッファ406の出力に結合されている第2の入力を有するMLU1610が含まれている。MLU1610の出力はその第2の入力に結合されているフィードバックループ1612を有する。演算論理装置(ALU)1614は第2の入力バッファ406の出力に結合されている第1の入力を有する。ALU1614はさらにMLU1610の出力に結合されている第2の入力を有する。ALU1614の出力はライティングモジュール54の出力に結合されている。ALU1614の出力と第3の入力バッファ408の出力はマルチプレクサ1616によりライティングモジュール54の出力に結合されていることに注意すべきである。

【0124】

次に、ALU1614の出力に結合されている入力と、ALU1614の第1の入力に結合されている出力とを有する第1のレジスタ装置1618が設けられている。第2のレジスタ装置1620はALU1614の出力に結合されている入力を有する。またこのような第2のレジスタ1620はMLU1610の第1の入力と第2の入力に結合されている出力を有する。

【0125】

ライティング論理装置（LLU）1622もまた設けられ、ALU1614の出力に結合されている第1の入力と、第1の入力バッファ404の出力に結合されている第2の入力と、MLU1610の第1の入力に結合されている出力とを有する。LLU1622の第2の入力は遅延素子1624により第1の入力バッファ404の出力に結合されていることに注意すべきである。さらに、LLU1622の出力は先入れ先出しレジスタ装置1626を介してMLU1610の第1の入力に結合されている。図16に示されているように、LLU1622の出力はまた変換モジュール1628によりMLU1610の第1の入力にも結合されている。動作において、このような変換モジュール1628は変換モジュール52と類似の方法でスカラー頂点データをベクトル頂点データへ変換するように構成されている。

【0126】

最後に、メモリ1500はMLU1610の入力と演算論理装置1614の出力の少なくとも一方に結合されている。特に、メモリ1500はMLU1610の第1および第2の入力に結合されている読み取り端子を有する。さらにメモリ1500はALU1614の出力に結合されている書き込み端子を有する。

【0127】

メモリは頂点データを処理するため、入力バッファ402、MLU1610、ALU1614、第1のレジスタ装置1618、第2のレジスタ装置1620、LLU1622と共に使用される複数の定数および変数を記憶している。

【0128】

図17は本発明の1実施形態による図16のライティングモジュール54のMLU1610の概略図である。示されているように、ライティングモジュール54のMLU1610は並列している3つの乗算器1700を含んでいる。動作において、本発明のMLU1610は2対3コンポーネントベクトルを乗算し、または1対3コンポーネントベクトルを通過するように構成されている。3コンポーネントベクトルの乗算はドット積または並列乗算により行われる。表13はライティングモジュール54のMLU1610が実行できる動作を示している。

【0129】

表13

【数6】

ZMLU_MULT	$o[0] = a[0]*b[0], o[1] = a[1]*b[1], o[2] = a[2]*b[2]$
ZMLU_PASA	$o[0] = a[0], o[1] = a[1], o[2] = a[2]$
ZMLU_PASB	$o[0] = b[0], o[1] = b[1], o[2] = b[2]$

【0130】

表14はライティングモジュール54のM L U1610の可能なAおよびB入力を示している。

表14

MA_V	V B U F F E R
MA_L	L L U
MA_R	R L U [2, 3] (MB_Rと共有)
MA_C	コンテキストメモリ (MB_Cと共有)
MB_M	M L U
MB_W	W B U F F E R
MB_R	R L U [2, 3] (MA_Rと共有)
MB_C	コンテキストメモリ (MA_Cと共有)

【0131】

図18は本発明の1実施形態による図16のライティングモジュール54のA L U1614の概略図である。示されているように、A L U1614は並列／直列の3つの加算器1800を含んでいる。使用において、A L U1614は2対3コンポーネントベクトルを加算し、または1対3コンポーネントベクトルを通過するように構成されている。表15はライティングモジュール54のA L U1614が実行できる種々の動作を示している。

【0132】

表15

【数7】

ZALU_ADD	$o[0] = a[0] + b[0], o[1] = a[1] + b[1], o[2] = a[2] + b[2]$
ZALU_SUM3B	$o[0] = a[0] + b[0] + b[1] + b[2]$
ZALU_PASA	$o[0] = a[0], o[1] = a[1], o[2] = a[2]$
ZALU_PASB	$o[0] = b[0], o[1] = b[1], o[2] = b[2]$

【0133】

表16はライティングモジュール54のALU1614の可能なAおよびB入力を示している。

表16

AA_W	W B U F F E R
AA_R	R L U [0, 1]
AB_M	M L U

【0134】

図19は本発明の1実施形態による図16のライティングモジュール54のレジスタ装置1618と1620の概略図である。示されているように、レジスタ装置1618と1620はそれぞれ2セットのレジスタ1900を含んでおり、レジスタ1900はそれぞれ対応するマルチプレクサ1902の第1の入力に接続されている出力と、マルチプレクサ1902の第2の入力に結合されている入力とを有する。

【0135】

ライティングモジュール54のレジスタ装置1618と1620はALU1614の2つのレジスタと、MLU1610の2つのレジスタに分離される。1実施形態ではこれらのレジスタはスレッドされている。レジスタ装置1618と1620は書き込みアドレスが読み取りアドレスと同一であるとき、入力から出力へのバイパス通路のためにゼロの待ち時間と有する。

【0136】

図20は本発明の1実施形態による図16のライティングモジュール54のLLU1622の概略図である。LLU1622はライティングモジュール54のライティング装置である。スカラーブロックは後に光+マテリアルカラーを乗算するために使用されるライティング係数を計算する。LLU1622は2つのMACと、インバータと、4つの小さいメモリとフラグレジスタを含んでいる。

【0137】

フラグレジスタはライティング方程式の条件付き部分を実行するために使用される。出力は環境、拡散、スペキュラー係数である。スカラーメモリはスペキュラー近似に使用される変数と定数を含んでいる。各メモリの第1の位置は (ctx0とctx2では) 1.0 および (ctx1とctx3では) 0.0 を含んでいる。1実施形態ではこれらはハードウェイアで結線され、ロードされる必要はない。

【0138】

使用において、LLU1622は機能的に式 $(x + L) / (M * x + N)$ を実行する。この式はスペキュラーライティング項を近似するために使用される。LLU1622への入力はライティングモジュール54のALU1614からあり、ライティング方程式で使用されるドット積である。図16に関して前述したように、LLU1622とMLU1610との間に出力 FIFO1626が存在し、これはMLU1610が係数を必要とするまで、係数をバッファする。1実施形態ではこのようなFIFO1626は遅延素子1608および1624、レジスタ1618および1620と共にスレッドされる。可能なカラーのマテリアル処理により、拡散およびスペキュラー出力がMLU1610により消費されるときはわからない。

【0139】

ライティングモジュール54はR, G, Bコンポーネントのみを処理するので、拡散出力アルファコンポーネントを処理するための特別に構成されたハードウェアが存在する。このような特別に構成されたハードウェアは2つのタイプのアルファコンポーネント、即ち vtx カラー ϕ [Tbuffeर] および記憶された ctx [ctx store] を出力できる。先のアルファコンポーネント間の選択はモードビット202により支配される。

【0140】

動作において、LLU1622はライティングの周囲 (Ca)、拡散 (Cd e)、スペキュラー (Cs) 係数を計算する。これらの係数は頂点のカラーに対する光の影響を生成するため周囲、拡散、スペキュラーカラーと乗算される。表16 AはLLU1622により受信された入力のリストと、ライティングの環境 (Ca)、拡散 (Cd e)、スペキュラー (Cs) 係数を生成するために実行される計算

を含んでいる。任意の所望のハードウェア構成は L L U1622の構成に使用されることに注意する。1 実施形態では、図20で示されている特別な構成が使用される。

【0141】

表16A

入力規定：

n = 正規ベクトル (変換エンジンから)
 e = 正規化されたアイベクトル (変換エンジンから)
 l = 正規化された光線ベクトル (変換エンジンから)
 s = スポットライトベクトル * 光線ベクトル (変換エンジンから)
 D = 距離ベクトル (l , d , $d * d$) (変換エンジンから)
 h = 半角ベクトル (変換エンジンから)
 K = 減衰定数ベクトル (K_0 , K_1 , K_2) (変換エンジンから)

L L U はその計算を実行するため以下のスカラーデータを受信する。

$n * 1$ (MLU / ALU から)
 $n * h$ (MLU / ALU から)
 $K * D$ (MLU / ALU から)
 s (変換エンジンから)
 パワー0 (ctx0-3メモリからのマテリアル指数)
 パワー1 (ctx0-3メモリからのスポットライト指数)
 距離 (ctx0-3メモリから)
 カットオフ (ctx0-3メモリから)

無限大光

L L U 計算：

$C_a = 1.0$

$C_d = n * 1$

$C_s = (n * h) ^ \text{power0}$

ローカル光

L L U 計算：

$a_{tt} = 1.0 / (K * D)$
 $C_a = a_{tt}$
 $C_d = a_{tt} * (n * 1)$
 $C_s = a_{tt} * ((n * h) ^ \text{power0})$

スポットライト

LLU計算：

$a_{tt} = (s ^ \text{power1}) / (K * D)$
 $C_a = a_{tt}$
 $C_d = a_{tt} * (n * 1)$
 $C_s = a_{tt} * ((n * h) ^ \text{power0})$

【0142】

前述したように、頂点シーケンサを制御するモードビットは頂点データ自体または頂点データから得られた結果により必ずしも変更されない。頂点データが頂点処理を変更することを可能にするため、LLU1622は与えられたフラグレジスタ1623を使用する。ビットをこのフラグレジスタでTRUEに設定することにより、フラグが計算の出力制御で特定されるならば、計算結果の0.0にクランプすることが可能である。フラグレジスタ1623の別の使用はレジスタ書き込みのための書き込みマスクを設定することである。

【0143】

フラグレジスタ1623は性能のペナルティがなくライティング方程式で0.0ヘイフ/ゼン/エルスクランピングを行うためにLLU1622中に設けられる。種々のオペランドの符号ビットはフラグを設定する。表16Bはフラグレジスタ1623のフラグが設定される方法と結果的なクランピングを示している。

【0144】

表16B

無限光

LLU計算：

$D_{flag} = (n * 1)$ のサインビット
 $S_{flag} = (n * h)$ のサインビット

クランプ：

```
C a = (0           ) ? 0 : C a ;
C d = (D f l a g ) ? 0 : C d ;
C x = (D f l a g | S f l a g) ? 0 : C s ;
```

局部光

LLU計算：

```
R f l a g = (range-d) のサインビット
D f l a g = (n * 1) のサインビット
S f l a g = (n * h) のサインビット
```

クランプ：

```
C a = (R f l a g           ) ? 0 : C a ;
C d = (R f l a g | D f l a g ) ? 0 : C d ;
C x = (R f l a g | D f l a g | S f l a g) ? 0 : C s ;
```

スポットライト

LLU計算：

```
C f l a g = (s-cutoff) のサインビット
R f l a g = (range-d) のサインビット
D f l a g = (n * 1) のサインビット
S f l a g = (n * h) のサインビット
```

クランプ：

```
C a = (C f l a g | R f l a g           ) ? 0 : C a ;
C d = (C f l a g | R f l a g | D f l a g ) ? 0 : C d ;
C x = (C f l a g | R f l a g | D f l a g | S f l a g) ? 0 : C s ;
```

【0145】

図21は本発明の1実施形態による図16のライティングモジュールに関連したフラグレジスタ1623の組織を示している。フラグレジスタ1623は8つの1ビットフラグを含み、ALU (FLAG) またはMACO (MFLAG) 出力の符号ビットにより設定される。

【0146】

L L U1622が3ワードにスメアされる場合M L U1610へスカラ値を出力するとき、フラグレジスタのマスクを特定する。レジスタとマスクが真であるならば、0, 0は出力を置換える。表17は出力された環境、拡散、スペキュラー属性で使用される図21の種々のフラグを示している。

表17

周囲マスク：	C, R,	U
拡散マスク：	D, C, R,	U
スペキュラーマスク：	D, S, C, R, T, U	

【0147】

スペキュラー項で使用される近似は実際の $\cos(\theta)^2$ が 0, 0 になる場合、負になる。結果として、クランピング動作を実行する必要がある。このため、T, U フラグが使用される。表18はL L U1622の機能論理装置 (F L U) 1621が行うことができる種々の動作を示している。図20に注意する。

【0148】

表18

Z F L U_I N V	$o = 1/a$ (仮数の正確度-12ビット)
Z F L U_I S Q	$o = 1/sqrt(a)$ (仮数の正確度-6ビット)
Z F L U_P A S S	$o = a$
Z F L U_P A S S 1	$o = 1.0$
Z F L U_M I N 1	$o = (a < 1.0) ? a : 1.0$
Z F L U_N O P	$o = 0.0$

【0149】

図22は本発明の1実施形態による図16のライティングモジュール54に関連したマイクロコードフィールドを示す図である。示されているように、ライティングモジュール54のマイクロコードは全体幅が85ビットである33フィールドに配置されている。フィールドは装置のデータ流を整合するように遅延される。M L U動作は遅延ゼロで行われ、A L U動作は遅延1で行われ、R L U、L L U出力動作は遅延2で行われる。各遅延は3サイクルに等しい。

【0150】

図23は本発明の1実施形態による図16のライティングモジュール54に関連したシーケンサ2300の概略図である。示されているように、ライティングモジュール54のシーケンサ2300はプロセス動作の複数のモードの状態を示すモードビット202を受信するように構成されている入力バッファ2302を含んでいる。また、それぞれモードの状態にしたがってプロセス動作を実行するように構成されているコードセグメントを記憶できるメモリ1502も含まれている。

【0151】

シーケンスモジュール2306はモードビットから得られる制御ベクトル2305に基づいてメモリ1502中の複数のアドレスを識別するためメモリ1502とバッファ2302との間に結合されている。シーケンスモジュール2306はさらに、ライティングモジュール54を動作するために使用されるコードセグメントを検索するためにメモリ1502中のアドレスをアクセスするように構成されている。

【0152】

ライティングモジュール54のシーケンサ2300は変換モジュール52のシーケンサと類似している。動作において、ライティングモジュール54のシーケンサ2300はスレッドされたモードビット202から得られるスレッドされた制御ベクトルによりステップし、それぞれのROMアドレスを実行し、その対応する制御ベクトルビットは“1”に設定される。制御ベクトルはROMが有するワードと同数のビットを有する。シーケンサ2300はスレッド毎に予め定められた数のサイクルで1つの“1”またはエネーブルビットのレートで任意の制御ベクトルをステップできる。モードビット202を使用しないコマンドはオンザフライマイクロコード発生により実行される。ライティングモジュール54のシーケンサ2300と変換モジュール52のシーケンサ1200との主な違いは、ライティングモジュール54のシーケンサ2300はループバックし8回までライティングコードを実行できることである。

【0153】

ライティングモジュール54のシーケンサ2300はそれぞれ新しい頂点ではゼロで開始し、マイクロコードシーケンスの終了時では1だけインクリメントする光カウンタを有する。モードビット202のLISフィールドが一致するビットフィ

ールドで“1”を含んでいるならば、シーケンサ2300は戻り、ライティングマイクロコードブロックの開始時でスタートする。これはゼロがL I Sフィールドで発見されるか、8つの光が行われるまで継続する。カラーの累算は拡散およびスペキュラーカラーを記憶するA L Uレジスタを（1光線毎に）インクリメントすることによって行われる。自動メモリアドレスのインデックスは各光線で正確なパラメータをフェッチするために光カウンタを使用して実行される。

【0154】

図24は本発明の1実施形態にしたがって変換モジュール52およびライティングモジュール54のシーケンサが関連したバッファの入力および出力を制御することができる方法について詳細に説明するフローチャートである。示されているように、頂点データは動作2420でバッファの第1のセットの1つのバッファで最初に受信される。頂点データが受信されるバッファはラウンドロビンシーケンスに基づいている。

【0155】

続いて、動作2422では、バッファの第2のセットのエンプティバッファもまたラウンドロビンシーケンスに基づいて識別される。変換モジュール52は第1のセットのバッファと、第2のセットのバッファとの間に結合されている。第2のセットのバッファのエンプティバッファが識別されると、頂点データは変換モジュールで処理され、変換モジュールから第2のセットのバッファの識別されたエンプティバッファへ出力される。動作ステップ2424および2426を参照。

【0156】

同様に、バッファの第3のセットのエンプティバッファまたはメモリ中のスロット或いはスペースは動作2428でラウンドロビンシーケンスに基づいて識別される。ライティングモジュール54はバッファの第2のセットと第3のセットの間に結合されている。バッファの第3のセットのエンプティバッファが識別されるとき、頂点データは動作2430で示されているようにライティングモジュールで処理される。頂点データはしたがってライティングモジュール52からバッファの第3のセットの識別されたエンプティバッファへ出力される。動作2432を参照。バッファまたはメモリ中のスロットの数はフレキシブルであり、変更されてもよい

ことに注意すべきである。

【0157】

図25は図24の方法にしたがって変換モジュール52およびライティングモジュール54のシーケンサが関連したバッファの入力および出力を制御することができる方法の説明図である。示されているように、第1のセットのバッファまたは入力バッファ400は変換モジュール52に出力を供給し、変換モジュール52は第2のセットのバッファまたは中間バッファ404、406に出力を与える。第2のセットのバッファ404、406はメモリ2550へ出力（ドレイン）するライティングモジュール54に出力を与える。

【0158】

図25で説明されている方法を実行するため、メモリ2550のスロットと、第1および第2のセットのバッファはそれぞれ頂点データを最初に受信したときに特有の識別子をそれぞれ割当てられる。さらに、各バッファの現在の状態は追跡される。このような状態は割当てられた状態、有効な状態、アクチブ状態または行われた状態を含んでいる。

【0159】

割当てられた状態は、バッファ／スロットが先のグラフィック処理モジュール、即ち変換モジュールまたはライティングモジュールの出力を受信するように既に割当てられていることを示している。書き込みポインタがラウンドロビンシーケンスでバッファ／スロットを走査しているとき、割当てられた状態のバッファ／スロットはこのような書き込みポインタを次のバッファまたはスロットにインクリメントさせる。

【0160】

バッファ／スロットが有効な状態であるならば、そのバッファ／スロットは頂点データを受信するために使用される。他方で、アクチブ状態はバッファ／スロットが現在、実行状態であるかまたは頂点データを受信していることを示す。このアクチブ状態はスレッドが完了するまで維持され、その後読み取りポインタをインクリメントし、したがってバッファ／スロットを有効状態に戻す。第1のセットのバッファ400はそれらを割当てるグラフィック処理モジュールが先に存在

しないので、単に有効状態であることだけができることに注意する。

【0161】

状態のシーケンスの1例を説明する。第1のセットのバッファ400と新しいコマンドビットのセット200の一方で頂点データを受信するとき、このようなバッファは有効状態に置かれ、その後バッファ402、404の第2のセットの1つが変換モジュール52の出力の予測において割当てられた状態に置かれる。

【0162】

バッファ404、406の第2のセットが割当に使用可能ではないならば、第1のセットのバッファ400中の頂点データは処理されることがない。さらに実行されるコードセグメントが同時に行われる他のコードセグメントと干渉するか否かを決定するためのチェックが行われる。干渉するならば、第1のセットのバッファ400の頂点データは処理されずストール（機能停止）状態が開始される。

【0163】

第2のセットのバッファ404、406の1つが割当状態に置かれた後、第1のセットのバッファ400はアクティブ状態に置かれる。変換モジュール52が実行を終了したとき、第2のセットのバッファ404、406は読み取られ、その後有効状態に置かれる。これらの状態の変化は第2のセット404、406とメモリ2550のスロット間の頂点データの転送中も同様に行われる。

【0164】

図25Bは設定モジュール57とトラバーサルモジュール58とを含むラスター化モジュール56を示している。ラスター化モジュール56は代わりの方法でエリアベースのラスター化を実行するように構成されている。特に、複数の多角形を規定するセンスポイントがプリミティブに、またはその近くに位置され、その後一次方程式がプリミティブ中に存在する画素を決定するためにそのポイントにおいて評価される。動作中、この評価はポイントが効率的な目的で代わりの方法で移動されるときに反復される。さらに、ラスター化モジュール56は何等クリッピングプロセスなしで動作するように構成される。

【0165】

図26はラスター化モジュール56の設定モジュール57の概略図である。示さ

れているように、設定モジュール57は所望の浮動小数点計算を実行するためにデータと制御信号をそれらの適切な機能装置へ導く処理をする制御セクション61を含んでいる。プリミティブシーケンサ62は頂点のシーケンスを三角形、直線または点に変える処理をする。さらに浮動小数点データパスセクション64は設定装置で必要とされる数学を実行するマルチプレクサおよび浮動小数点計算装置を含んでいる。

【0166】

図26の参照を続けると、ラスター化装置は整数値でのみ動作するので、出力フォーマットセクション63はエッジスロープとエッジ値の内部浮動小数点フォーマットをラスター化装置に適している整数のフォーマットに変換する処理をする。勿論、別の実施形態では、ラスター化装置は浮動小数点を使用し、したがって出力フォーマットセクション63の必要性をなくすことができる。

【0167】

動作において、出力フォーマットセクション63はブロック浮動小数点変換を実行する。よく知られているように、所定の数、即ち 2.34×10^0 では、浮動小数点フォーマットは仮数 (2.34) とその指数 (10) を追跡する。ブロック浮動小数点変換は基本的に指数が同一であるように、入来するデータの仮数の小数点位置を操作する。このため、指数はラスター化モジュール56で処理される必要はない。

【0168】

図26Aは図25Bのラスター化モジュール56の設定モジュール57によって計算される種々のパラメータを示している。このようなパラメータは関連する機能を実行するためにラスター化モジュール56に必要とされる。プリミティブ2600を受信するとき、設定モジュール57はプリミティブ2600のスロープ2601、スタート位置2602、スタート値2604を含む3つの値を計算する。

【0169】

スロープ2601はラスター化中に使用されるプリミティブ2600のエッジの一次方程式の係数を生成するために使用される。スロープ2601は例えば以下示す式#4および#5を使用することにより計算される。

式#4および#5

$$\text{スロープ}_A = y_0 - y_1$$

$$\text{スロープ}_B = x_1 - x_0$$

ここで y_0 、 y_1 および x_0 、 x_1 は図26Aで示されている頂点の座標である。

【0170】

スロープはまた1つの回転動作等を使用することによって頂点の座標を使用して計算されることに注意する。

【0171】

スタート位置2602はさらに以下詳細に説明するようにエリアラスター化のスタート点を示している。スタート値2604は図26Aで示されている陰影を付けられた三角形の面積に等しく、またエリアベースのラスター化プロセス中にも使用される。このようなスタート値2604はスクリーンについてのラスター位置をステップするように選択され、各ステップでスロープを付加することはラスター位置がエッジにあるとき丁度ゼロに等しい。スタート値2604の計算は以下の式#6を使用して実現される。

式#6

$$\text{starting_value} = \text{スロープ}_A * (x_s - x_0) + \text{スロープ}_B * (y_s - y_0)$$

ここで、 x_s 、 y_s = スタート位置2602、

スロープ_A、スロープ_B = 図26Aで示されている座標に基づいた1つのエッジのスロープ、

x_0 、 y_0 = 図26Aで示されているエッジの頂点の1つの座標

【0172】

前述の値はまた他のタイプのプリミティブに対して計算されることを理解すべきである。例えば、直線の場合、余分のスロープは4つの側面の境界のあるボックスで計算されなければならない。このようなスロープは境界のあるボックスの反対側のスロープの逆数を取ることにより容易に計算されることができる。余分のスロープの計算に加えて、別のスタート値が直線のプリミティブの場合に計算されることを必要とすることに注意すべきである。

【0173】

図27はラスター化モジュール56が例えれば三角形等の複数のプリミティブのうちの1つを処理する方法を示している。特に、最初の動作は最初にラスター化装置のモジュール56の設定モジュール57により実行される。プリミティブを受信するとき、一次方程式の一次方程式係数は当業者によく知られた方法で図26Aのスロープ2601を使用して動作2700でプリミティブを規定する直線で決定される。よく知られているように、3つの一次方程式が三角形を規定するのに必要とされる。他方で、直線のようなプリミティブは4つの側面と4つの一次方程式により長方形または平行四辺形として描かれる。

【0174】

その後、動作2702では、任意のプリミティブ頂点が負のW一座標を有するならば、その一次方程式係数は変更される。このプロセスに関する付加的な情報を図32を参照してさらに詳細に説明する。

【0175】

ラスター化モジュール56の設定モジュール57もまたプリミティブの境界のあるボックスを計算することに注意しなければならない。ほとんどの三角形では、境界を有するボックスは3つの頂点の最小値および最大値を含んでいる。直線では、境界を有するボックスの4つの平行四辺形のコーナーが計算される。負のW一座標の頂点を有する三角形または直線では、描かれるエリアは頂点の凸閉の殻を超えて延在する。

【0176】

OpenGL（商標名）のコマンドの1つは描かれない境界外を規定するシザーロング方形である。ラスター化モジュール56の設定モジュール57は境界のあるボックスとシザーロング方形との交差点を計算する。シザーロング方形は長方形であるので、4つの付加的な一次方程式が与えられる。シザーロング方形に関連する一次方程式は平凡な形状、即ち水平または垂直を有することに注意する。

【0177】

さらに、3-Dスペースでは、近距離の平面と遠距離の平面とは平行であり、視線に対して直角である。プリミティブが三角形である場合、3つの頂点が含

まれ、任意の方位を有する平面を規定する。プリミティブの平面と、近距離および遠距離の平面との交差点は2つの関連する一次方程式を有する2つの直線を含んでいる。

【0178】

したがって、各プリミティブはそれが三角形または直線の形態を取るかに応じて全部で9または10の一次方程式をそれぞれ有する。再び三角形の場合、このような一次方程式は三角形を規定する3つの一次方程式と、境界のあるボックスを規定する4つの一次方程式と、プリミティブが存在する平面と近距離の平面および遠距離の平面との交差点を規定する2つの一次方程式とを含んでいる。

【0179】

図27を参照し続けると、プロセスは動作2704で進行し、プリミティブ上またはその近くの複数の点を位置付ける。スタート位置2602は図26Aで示されているように、このような位置付けを指示している。このような点は含まれる凸形領域を規定し、凸形領域のコーナーに位置している。図27Aは例えば長方形等の凸形領域2707を囲むこのようなセンスポイント2705を示している。1実施形態では、このような長方形はサイズが8×2画素である。さらに点はプリミティブの上部の頂点を囲むように最初に位置される。選択肢として、これは切捨てを使用して実現されてもよい。

【0180】

プリミティブが一度位置付けられると、プロセスは以下説明する方法でプリミティブの行を処理することにより動作2706で開始するトラバーサルモジュール58により継続される。各行の処理後、ジャンプ位置が決定2708で発見されるか否かを決定する。ジャンプ位置は次の行を処理するためスタート位置にあり、以下詳細に説明する。決定2708でジャンプ位置が発見されたことが決定されるならば、凸面領域を規定するセンスポイントは動作2710に移動される。しかしながら、ジャンプ位置が発見されていないことが決定されたならば、プロセスは終了される。別の実施形態では、列、対角線または任意の他のタイプのストリングが行の代わりに動作2706で処理されることに注意すべきである。

【0181】

図28は図27の処理行動2706に関連した本発明のプロセスを示すフローチャートである。示されているように、プロセスは多角形を規定するセンスポイントが決定2801で右に移動されるか否かを決定するため、動作2800でセンスポイントを計算することにより開始する。このような決定は最も右のセンスポイントの位置に基づいて行われる。最も右のセンスポイントがプリミティブの同一エッジ外に位置されないならば、右方向の移動は許容され、現在位置の右への位置（XおよびY座標）は動作2802でスナップ位置として記憶される。しかしながら、最も右のセンスポイントがプリミティブの1以上のエッジ外に位置されるならば、右方向の移動は許容されず、動作2802はスキップされる。

【0182】

次に、一次方程式は動作2804で凸形領域、例えば長方形の点で評価される。この評価は点がプリミティブ中に存在するか否かの決定を含んでいる。ポイントがプリミティブ中に存在するか否かについてのこのような決定は、各一次方程式の評価が各センスポイントで正の値または負の値を与えるか否かを決定することを含んでいる。

【0183】

一次方程式はプリミティブ内では正であり、その外では負であるように公式化することができる。画素が丁度エッジ上に存在する包含的なエッジが描かれ、ゼロに評価され、正として扱われる。描かれるべきではない排他的なエッジは開始の一次方程式の値から1の値を最初に減算することにより負にされることができる。したがって、排他的エッジ上の画素は正のゼロの代わりに負値（-1）に評価される。これはセンスポイントの移行が包含的／排他的ポリシーを無視し、単に一次方程式の符号を試験することを許容する。

【0184】

一次方程式が点において評価された後、決定2806でセンスポイントの現在の位置がジャンプ位置を構成するか否かが決定される。2つの下部のセンスポイントが両者ともエッジ外でなければ、ジャンプ位置は記憶されることに注意すべきである。決定2806で、ジャンプ位置が発見されたことが決定されたならば、動作2808でジャンプ位置が計算され記憶される（または存在するならば先に記憶され

たジャンプ位置で置換する）。しかしながらノーであるならば、動作2808はスキップされる。

【0185】

図28の参照を続けると、決定2810で、最も左のセンスポイントが両者ともプリミティブのエッジ外であるか否かが決定される。このプロセスは再び両者の最も左のセンスポイントの一次方程式の評価が正または負値を与えるか否かを決定することを含んでいる。特に適切なセンスポイントで9または10のエッジ式の係数を計算するとき、9または10値が与えられ、それらは9または10の符号ビットを有する。現在の側面が完全にエッジ外であるか否かを決定するために、例えば本発明は2つのセンスポイントからの10の符号ビットを共に論理積（AND）処理する。任意のビットが残存するならば、両者のポイントはそのエッジ外である。

【0186】

最も左のセンスポイントが両者ともプリミティブエッジ外ではないことが決定されたならば、左方向にあると考えられるプリミティブの部分がさらに残留していることが結論付けされ。センスポイントは動作2812に左へ移動される。決定2810で、両者の最も左のセンスポイントがプリミティブのエッジ外であることが決定されたならば。左方向にあると考えられるプリミティブの部分がさらに残留していないことが結論付けされる。次に、決定2814で、動作2802から得られたスナップ位置が存在するか否かの決定が行われる。

【0187】

決定2814で、スナップ位置が存在しないことが決定されたならば、プロセスは行われる。しかしながら、スナップ位置が存在するならば、センスポイントは動作2816でスナップ位置に移動される。その後、2804-2812の動作に類似した動作はプリミティブの右側をマップするように行わされる。これは凸形領域の点で一次方程式を評価することにより動作2818で開始する。

【0188】

一次方程式が点で評価された後、センスポイントの現在の位置が動作2820でジャンプ位置を構成するか否かが決定される。決定2820で、ジャンプ位置が発見

されたことが決定されたならば、動作2822でジャンプ位置が計算され記憶される。ノーであるならば、動作2822はスキップされる。

【0189】

図28の参照を続けると、決定2824で、最も右のセンスポイントが両者ともプリミティブのエッジ外であるか否かが決定される。最も右のセンスポイントが両者ともプリミティブのエッジ外ではないことが決定されたならば、右方向にあると考えられるプリミティブ部分がさらに残留していることが結論付けされ、センスポイントは動作2826で右に移動される、決定2824で、最も右のセンスポイントが両者ともプリミティブのエッジ外にあることが決定されたならば、右方向にあると考えられるプリミティブの部分がさらに残留していないことが結論付けされ、瞬時のプロセスが実行される。

【0190】

図28Aと図28Bは本発明のセンスポイントがプリミティブ2850に関して移動されるシーケンスを示している。種々の代りの方法が決定2800で点が左に移動することができるか否かを決定し、最初に右に進行することを含むことに注意する。一次方程式は点が任意の所望の方法でプリミティブ内または外であるかを示すために規定される。

【0191】

反復するループのステップ動作を防ぐため、本発明はしたがってラスター化中に全体的な移動方向を使用する。最初の構成はトップーダウンを行い、次へステップダウンする前に1つの行の1つ1つの凸領域に行く。行のトップーダウンを行い、右その後左へ、または左その後右へステップしないことによりループは阻止される。

【0192】

前述のプロセスの1例は図27Aの多角形を規定する点P1、P2、P3、P4を参照して示されている、動作において、隣接するセンスポイントの対はそれらの方向のステッピングが生産的 (productive) であるか否かを決定するため検査ができる。例えば図27A中のP3とP4の両者が多角形のエッジ外であるが、P1および／またはP2は多角形のエッジ外ではないならば、明

白に描くことのできる内部エリアは右ではなく左に位置する。したがってセンスポイントは右へ移動すべきではない。反対に、P3とP4の両者が全てのエッジ内であるならば、描くことのできる内部エリアは丁度P3とP4を越えて存在し、右へのステップが適切である。P3とP4が同じエッジの外ではないならば、右へのステップが生産的である。この同じ論理はP1とP3により誘導される上方向へのステップまたは、P1とP2により誘導される左のステップ、またはP2とP4に基づいた下方向のステップにも適用される。

【0193】

前述のプロセスはしたがってガイドとしてセンスポイントを使用して、プリミティブの内部周辺の点により規定される凸形区域を移動またはステップする。点によって規定される凸形領域が大きいので、多数の画素は同時に試験される。使用中、全てのセンスポイントがプリミティブの全てのエッジ内であるならば、全ての囲まれた画素は描かれることが可能でなければならない（凸形のプリミティブを想定する）。コーナー部を検査することにより多くの利点が与えられ、即ちプリミティブの任意のエリアを与える能力は内部、外部または分割である。後者のケースでのみ、点により規定される凸形領域の個々の画素が試験される必要がある。このような場合、点により規定される凸形領域の画素はこれらがプリミティブに存在するか否かを決定するために別の方法により1つづつ試験される。さらに、センスポイントはエリアを分割するエッジと分割しないエッジを規定することにより、必要な試験の量を減少する。

【0194】

図29は図27の処理行動作2706に関連した本発明のプロセスの別の犁耕体プロセスを示すフローチャートである。示されているように、最初に決定2900で、先の移動が第1または第2の方向であるかを決定する。実際に先の移動が存在しなかつたならば、デフォルトの先の移動が仮定される。決定2900で、先の移動が第2の方向であることが決定されたならば、動作2902で図28の動作2804と類似した方法で一次方程式が凸形領域、例えば長方形の点で評価される。

【0195】

図29の参照を続けると、次に、決定2904で、長方形の第1の側面のセンス

ポイントが両者ともプリミティブのエッジ外であるか否かに関して決定が行われる。ノーであるならば、センスポイントは動作2906で第1の方向で移動またはステップされる。長方形の第1の側面のセンスポイントが両者ともプリミティブのエッジ外であるという決定が行われると、決定2905で、点が下方向に移動できるか否か、換言すると、現在位置がジャンプ位置を構成するか否かが決定される。イエスならば、動作2908でジャンプ位置が計算され、記憶され、その後プロセスが行われる。

【0196】

他方で、決定2900で、先の移動が第1の方向であることが決定されたならば、動作2902-2908と類似の動作が実行される、特に動作2910で、動作一次方程式は凸形領域、例えば長方形の点で評価される。決定2912で、長方形の第2の側面のセンスポイントが両者ともプリミティブのエッジ外であるか否かに関する決定が行われる。ノーであるならば、センスポイントは動作2914で第2の方向で移動またはステップされる。長方形の第2の側面のセンスポイントが両者ともプリミティブのエッジ外であるという決定が行われると、決定2913で、点が下方向に移動できるか否か、換言すると、現在位置がジャンプ位置を構成するか否かが決定される。イエスならば、動作2916でジャンプ位置が計算され、記憶され、その後プロセスが行われる。

【0197】

図29Aは図29の犁耕体プロセスにしたがって本発明のセンスポイントがプリミティブに関して移動されるシーケンスを示している。前述の犁耕体ラスター化はハードウェアに対してより良好な性能を与えるあるルールに従うようにシーケンスを規制する。示されているように、犁耕体ラスター化は前後に曲がる蛇行パターンを与える。水平の犁耕体シーケンスは例えばプリミティブ三角形内に全ての画素を生成し、それらは左から右へ1つの行に存在し、その後、次の行で右から左へ画素を生成する。このような折曲がったパスは生成された画素から最近予め発生された画素までの平均距離が比較的小さいことを確実にする。

【0198】

ほぼ最近に予め発生された画素の発生は、画素および/またはそれらの対応

するテキスチャ値が限定されたサイズのメモリ中に維持されるときに重要である。犁耕体シーケンスはこのようなメモリに既にロードされている画素またはテキスチャを頻繁に発見し、それ故メモリのロードの反復が行われる頻度が少なくなる。

【0199】

1つの選択肢として、ラスター化の前にプリミティブを複数の部分に分離する少なくとも1つの境界が使用される。動作において、点は各部分で別々に移動される。さらに、点は第2の部分で移動される前に第1の部分の全体を移動される。

【0200】

図30は境界を使用する別の犁耕体プロセスを示しているフローチャートである。1つの選択肢として、境界を使用するか否かの決定はプリミティブの大きさに基づく。図30で示されているように、境界を処理する犁耕体プロセスは、少なくとも1つの境界が規定され、プリミティブを複数の部分またはスワス(swath)に分割する付加的な動作3000を除いて図27のプロセスと類似している。

【0201】

図30の参照を続けると、付加的な決定3001はプリミティブのあらゆる部分の完了にしたがう。特に、決定3001で、隣接部分のスタート位置が動作3006で発見されるか否かが決定される。イエスであるならば、センスポイントにより規定される凸形領域は動作3002でプリミティブの隣接部分のスタート点に移動され、動作3004-3010はプリミティブの新しい部分に対して反復される。さらに動作3006のスタート位置の決定に関する情報を図31を参照してさらに詳細に説明する。

【0202】

図31Aは図30の境界ベースの犁耕体プロセスにしたがって本発明の凸形状領域がプリミティブに関して移動されるプロセスを示している。示されているように、処理される第1の部分はプリミティブの最上部の頂点を含む部分である。動作中、左の隣接部分が処理され、その後近接する左の隣接部分が処理され、以下同様に処理される。これは、左の隣接部分がなくなるまで継続される。次に

第1の部分の右への隣接部分が処理され、その後、近接する右の隣接部分が処理され、全ての右の隣接部分が処理されるまで継続される。他のタイプの順序付け方式がユーザの要望にしたがって使用されてもよいことを認識すべきである。

【0203】

図31は図30のプロセス行の動作3006に関連したプロセスを示すフローチャートである。このようなプロセスは決定3118と3121を除いて図29の犁耕体プロセスに類似している。決定3118と3120の両者は任意のセンスポイントが任意の境界を通過しているか否かを決定する。センスポイントが境界内であることが決定されさえすれば、それぞれのループが継続される。

【0204】

動作3119と3121では、プリミティブの隣接部分のスタート位置は検索され、決定3118と3120で、凸形領域の任意のセンスポイントがそれぞれ任意の境界を通過していることを決定したときに記憶される。図31Aで示されているように、このようなスタート位置3126は境界を越えて存在するプリミティブ部分の最上部点としてそれぞれ規定される。この位置を記憶することにより、プロセスがプリミティブにおける隣接する境界の規定された部分で反復されるときにスタート点が与えられる。

【0205】

動作3119と3121は両者ともプリミティブの第1の部分を処理しながら実行されることに注意する。図31で明白に示していないが、部分を第1の部分の左に処理するときこのような動作の第1の動作だけが行われ、部分を第1の部分の右に処理するとき、このような動作の第2の動作だけが行われる。換言すると、部分を第1の部分の左に処理するとき、スタート位置は現在処理された部分の最も左の境界が超過されたときだけ決定される。同様に、部分を第1の部分の右に処理するとき、スタート位置は現在処理された部分の最も右の境界が超過されたときだけ決定される。

【0206】

ラスター化中に境界を使用することは、パイプライン処理中の非常に臨界的な問題を解決する。プリミティブが非常に広いならば、1つの行の画素に関連す

る記憶媒体は限定されたサイズのメモリに適合しない。境界によるラスター化は三角形を限定された幅の行（または列）に分離し、次の部分へ移動する前に、このような部分内に全ての画素を生成する。

【0207】

例えば、三角形が100画素幅であっても、限定されたサイズの画素またはテキスチャメモリは先の20画素の情報だけを保持する。画素シーケンスを10画素幅の垂直部分内に存在するように制限することによって、以前のおよび現在の行の全ての画素はメモリに適合することが可能である。これは、境界の規定された部分内の犁耕体シーケンスが常にメモリの（存在するならば）現在の行の以前の画素と、メモリの（存在するならば）上の行の画素とを有することを意味している。

【0208】

ほとんどの基礎的なメモリシステムはブロック単位のあるオーバーヘッドによりデータのブロックを転送する。メモリシステムに対する小さいアクセスはこのオーバーヘッドにより重いペナルティを課される。効率的であるように、大きいアクセスが使用され、ブロックの残りは次に使用される場合のために維持される。さらに、キャッシュメモリシステムは複数のこれらの最近のブロックを維持し、メモリアクセスが避けられることができる確率を増加させる。

【0209】

本発明の犁耕体シーケンスは、現在のラインの1端部のすぐ下の画素を反転し処理するときにシングルリテインーブロックを使用する。さらに、犁耕体シーケンスはラスター化を特定サイズの部分に限定するときキャッシュを使用する。特に部分内の2つの走査線はキャッシュに適合され、第2の走査線を通じて第1の走査線のキャッシュ記憶から利点が得られる。

【0210】

シーケンスまたは境界の規定された部分の数には制限はない。本発明は垂直部分および水平の犁耕体パターンの例を使用したが、類似の原理が水平部分および垂直の犁耕体パターンまたは、対角線部分およびパターンまで拡張される。1実施形態では、ストリング（例えば、行、列、対角線等）の長さはストリングが

存在するプリミティブの大きさよりも小さいようにそれぞれ限定される。

【0211】

図32は図27の動作2702に関連したプロセスに関連したプロセスを示すフローチャートである。瞬間的なプロセスは目の後に存在する部分でプリミティブを処理するように設計されている。これらの域外の部分はその次のラスター化動作で問題を生じる。これを実現するため、瞬間的なプロセスは変数Wを使用し、これは投影、即ち遠近法でオブジェクトを観察するために共通して使用される。変数Wは他の座標X、Y、Zが近くのものを大きく、遠くのものを小さくするために割算される数字である。変数Wは投影の中心と、対応する頂点との間の距離を表す。

【0212】

図32で示されているように、プリミティブは最初に受信され、複数の頂点により規定される。そのような各頂点はW値を含んでいる。プリミティブを受信するとき、設定モジュールは頂点に基づいてプリミティブを特徴付けするラインを規定する役目を行う。動作3200に注意。

【0213】

W値はその後、決定3202で解析される。示されているように、1つのW一値が負であるならば、負の値を有する頂点と反対のラインの一次方程式は動作3204でフリップされる。換言すると、一次方程式の係数は-1により乗算される。さらに、2つのW一値が負であるならば、正のW一値を有する頂点と、負のW一値を有するそれぞれの頂点とを接続するラインの一次方程式は動作3206でフリップされる。3つのW一値が負であるならば、不合格（カル）状態3207が生じ、本発明は三角形を不合格とする。負であるW一値がないならば、付加的な措置は取られない。

【0214】

図32A-32Cはフリップ一次方程式が、処理されるスクリーンの部分に影響を与える方法を示している。図32AはW値が負のものではなく、一次方程式が変更されない状態の場合を示している。示されているように、プリミティブの内部部分はこのようなケースで満たされている。

【0215】

図32Bは1つのW一値が負であり、したがってその一次方程式がフリップされるケースを示している。示されているように、頂点と対向するプリミティブ部分は現在のケースで満たされている。特に、描かれるエリアは-W頂点を共有する2つの三角形の面と共直線性である2つのラインにより境界を与えられ、さらに、2つの+W頂点を共有する三角形の面により境界を与えられる。

【0216】

図32Cは2つのW一値が負であり、したがってその一次方程式がフリップされるケースを示している。示されているように、頂点と対向するプリミティブ部分は図27-32を参照して前述した方法および/またはプロセスを使用して満たされる。換言すると、描かれるエリアは+W頂点を共有する2つの三角形の面と共直線性である2つのラインにより境界を与えられ、さらに、+W頂点に接近する。

【0217】

本発明はしたがって全ての3つの前述のケースを処理することができる。三角形の部分が近距離および/遠距離の平面を越えているならば、これらの平面内にその部分だけを描く。三角形が1または2の負のZ頂点を有するならば、正確な+Z部分だけが描かれる。

【0218】

全ての頂点がオフスクリーンであり、三角形が目の後方から遠距離の平面を越えて延在しても、画素は三角形内およびスクリーン上であり、近限界と遠限界との間にZを有する。本発明は悪い画素を使用する時間の浪費を少なくすることを確実にする。スクリーンエッジまたは近距離平面と遠距離平面による全てのクリッピングが容易に使用されることができる凸形領域のオンスクリーンで常に行われるので、これは可能である。

【0219】

スタート点が満たされるエリア内ではないときに時によって問題が生じる。上部頂点がオフスクリーンであるか近距離の平面または遠距離の平面によりクリップされる場合にこれは生じる。この場合、トラバーサルステージは描かれる区

域の上部点を検索しなければならず、上から開始する。これは三角形のエッジスロープとZスロープの符号により誘導されることにより効率的に行われる。これは三角形の一次方程式が描かれる領域外であることとその理由を発見するために、三角形の一次方程式を試験できる。外部にあるエッジおよび／またはZ限界を知ったとき、そのエッジまたは限界へ近付けるステップ方向を知る。（選択肢のあるとき）好みによって水平から垂直に移動することによって、描かれた領域の検索は上部に描くことのできる画素が存在するならば、それを発見する。オープンアップする外部（-W）三角形でもこの問題は生じる。この場合、描かれる区域は全ての3つの頂点よりも上方に延在する。

【0220】

本発明の1実施形態ではトラバーサルは三角形の上部から下部へ進行する。負のW-値をもたず頂点がシザー長方形であるならば、スタート点は三角形の上部の頂点である。トラバーサルは常にシザー長方形内で開始し、その外ではないので、エッジにより囲まれるエリアがシザー長方形を越えて延在しても、シザー長方形内の三角形部分だけが描かれる。このようにして、簡単なシザー長方形の長方形エッジクリッピングが行われる。

【0221】

種々の実施形態を前述したが、これらは技術的範囲の限定ではなく例示でのみ示されていることを理解すべきである。したがって、本発明の技術的範囲は前述の例示的な実施形態により限定されず、特許請求の範囲とそれらの均等物にしたがってのみ限定される。

【図面の簡単な説明】

【図1】

従来技術のグラフィック処理方法のフロー図。

【図1A】

従来技術のスキニング方法を示す概略図。

【図1B】

単一の半導体プラットフォーム上に構成された本発明の1実施形態の種々のコンポーネントを示すフロー図。

【図2】

本発明の1実施形態による頂点属性バッファ（VAB）の概略図。

【図2A】

本発明の1実施形態によるVABによって受取られることのできる種々のコマンドを示すチャート。

【図2B】

本発明の1実施形態によるVABとの間で頂点属性をロードし、ドレインする方法を示すフローチャート。

【図2C】

図2Bの動作を実行するために使用される本発明のアーキテクチャの概略図。

【図3】

本発明の1実施形態によるVABに関連したモードビットの説明図。

【図4】

本発明の変換モジュールを示す概略図。

【図4A】

本発明の1実施形態による多数の実行スレッドを実行する方法を示すフローチャート。

【図4B】

本発明の1実施形態にしたがって図4Aの方法が行われる手順を示すフロー図。

【図5】

本発明の1実施形態による図4の変換モジュールの機能装置の概略図。

【図6】

図5の変換モジュールの乗算論理装置（MLU）の概略図。

【図7】

図5の変換モジュールの演算論理装置（ALU）の概略図。

【図8】

図5の変換モジュールのレジスタファイルの概略図。

【図9】

図5の変換モジュールの反転論理装置（I L U）の概略図。

【図10】

本発明の1実施形態による図5の変換モジュールの出力コンバータの出力アドレスのチャート。

【図11】

本発明の1実施形態による図5の変換モジュールのマイクロコード編成図。

【図12】

本発明の1実施形態による図5の変換モジュールのシーケンサの概略図。

【図13】

図12の変換モジュールのシーケンサの使用に関連した種々の動作を詳細に示すフローチャート。

【図14】

図12の変換モジュールのシーケンサのシーケンシングコンポーネントの動作を詳細に示すフロー図。

【図14 A】

グラフィック処理中にスカラーおよびベクトル成分を処理するために使用される本発明のコンポーネントを示すフロー図。

【図14 B】

図5の変換モジュールに対応している図14 Aに示されている本発明の機能コンポーネントの1つの可能な組合せ1451を示すフロー図。

【図14 C】

図14 Aに示されている本発明の機能コンポーネントの別の可能な組合せ1453を示すフロー図。

【図14 D】

本発明の1実施形態にしたがって図12の変換モジュールにより実施されるグラフィック処理中にブレンディング動作を行う方法を示すフロー図。

【図15】

本発明の1実施形態のライティングモジュールの概略図。

【図16】

本発明の1実施形態による図15のライティングモジュールの機能装置を示す概略図。

【図17】

本発明の1実施形態による図16のライティングモジュールの乗算論理装置（M L U）の概略図。

【図18】

本発明の1実施形態による図16のライティングモジュールの演算論理装置（A L U）の概略図。

【図19】

本発明の1実施形態による図16のライティングモジュールのレジスタ装置の概略図。

【図20】

本発明の1実施形態による図16のライティングモジュールのライティング論理装置（L L U）の概略図。

【図21】

本発明の1実施形態による図16のライティングモジュールに関連したフラッグレジスタの説明図。

【図22】

本発明の1実施形態による図16のライティングモジュールに関連したマイクロコードフィールドの説明図。

【図23】

本発明の1実施形態による図16のライティングモジュールに関連したシーケンサの概略図。

【図24】

本発明の1実施形態にしたがって変換およびライティングモジュールのシーケンサが関連したバッファの入力および出力をどのように制御することができるかを詳細に説明するフローチャート。

【図25】

図24の方法にしたがって変換およびライティングモジュールのシーケンサが

関連したバッファの入力および出力をどのように制御することができるかを示す概略図。

【図25B】

図1Bのラスター化装置の種々のモジュールの概略図。

【図26】

本発明のラスター化モジュールの設定モジュールの概略図。

【図26A】

図26のラスター化装置の設定モジュールによって計算される種々のパラメータを示す説明図。

【図27】

図26に示されているラスター化装置コンポーネントの設定およびトラバーサルモジュールに関連した本発明の方法を示すフローチャート。

【図27A】

本発明の1実施形態にしたがってプリミティブにおけるエリアを識別するために移動される凸状領域を囲む方向ポイントを示す説明図。

【図28】

図27の処理行動2706に関連している本発明のプロセスを示すフローチャート。

【図28A】

本発明の凸状領域がプリミティブに関して移動されるシーケンスを示す概略図。

【図28B】

本発明の凸状領域がプリミティブに関して移動されるシーケンスの別の例を示す概略図。

【図29】

図27の処理行動2706に関連した本発明のプロセスの別の犁耕体プロセスを示すフローチャート。

【図29A】

図29の犁耕体プロセスにしたがって本発明の凸状領域がプリミティブに関し

て移動されるシーケンスを示す概略図。

【図30】

境界を使用する別の犁耕体プロセスを示すフローチャート。

【図31】

図30の動作3006に関連したプロセスを示すフローチャート。

【図31A】

図30および31の境界ベースの犁耕体プロセスにしたがって本発明の凸状領域がプリミティブに関して移動されるシーケンスを示す概略図。

【図32】

図27の動作2702に関連したプロセスを示すフローチャート。

【図32A】

図32のプロセスにおいて負のW値が1つも計算されないとき、どのようなエリアが描かれるかを示す説明図。

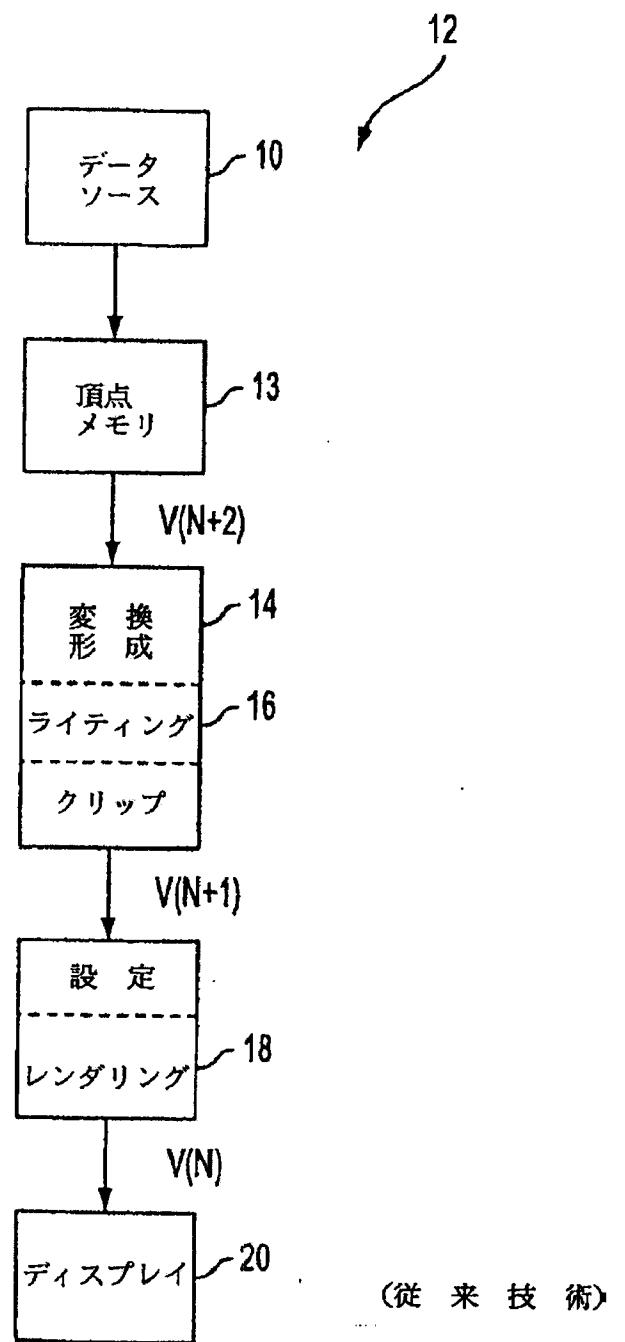
【図32B】

図32のプロセスにおいて負のW値が1つだけ計算されたとき、どのようなエリアが描かれるかを示す説明図。

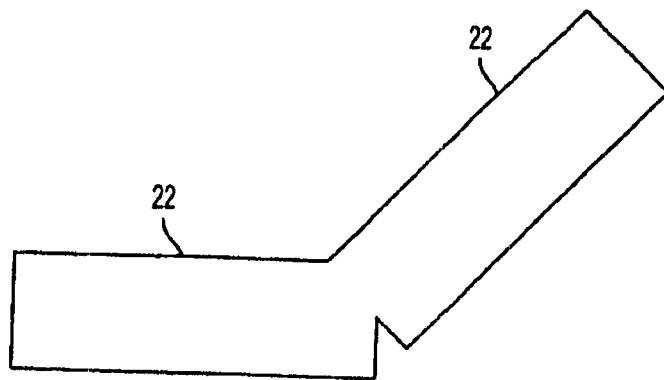
【図32C】

図32のプロセスにおいて負のW値が2つだけ計算されたとき、どのようなエリアが描かれるかを示す説明図。

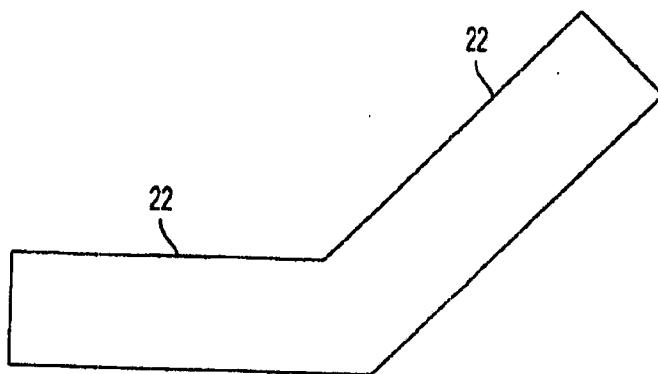
【図1】



【図1A】



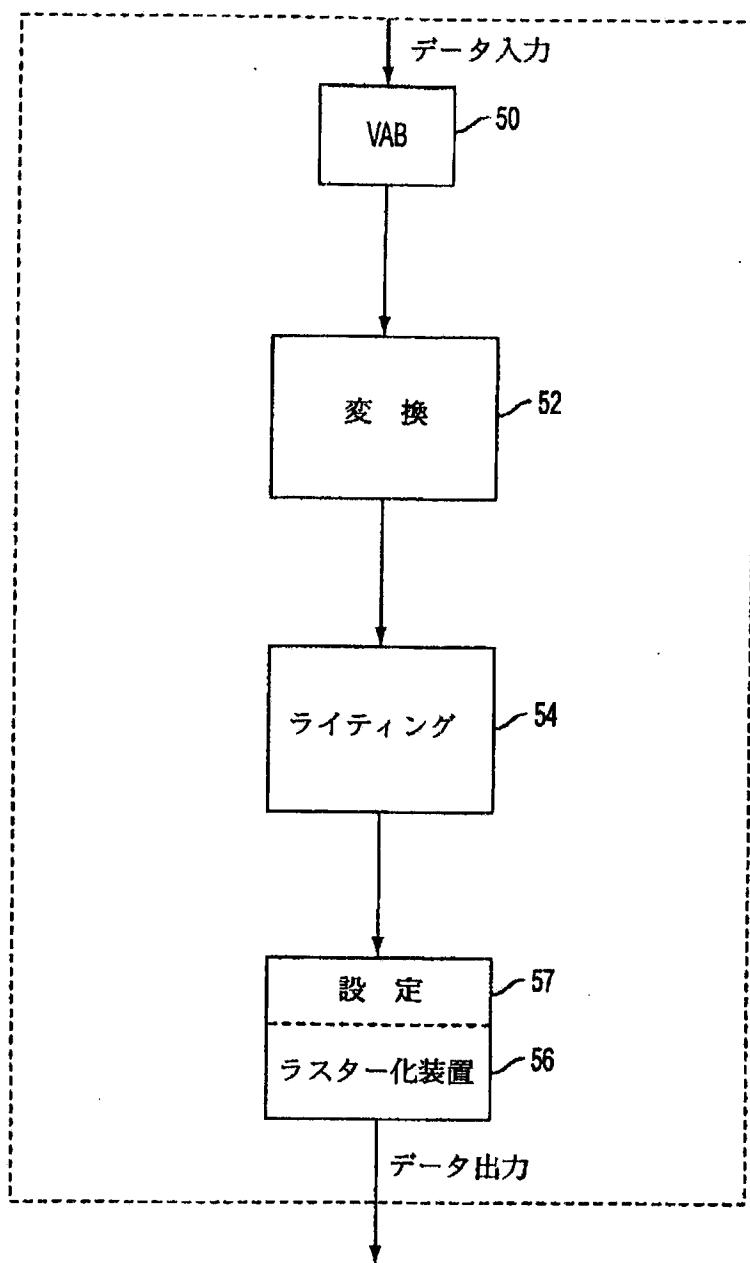
前



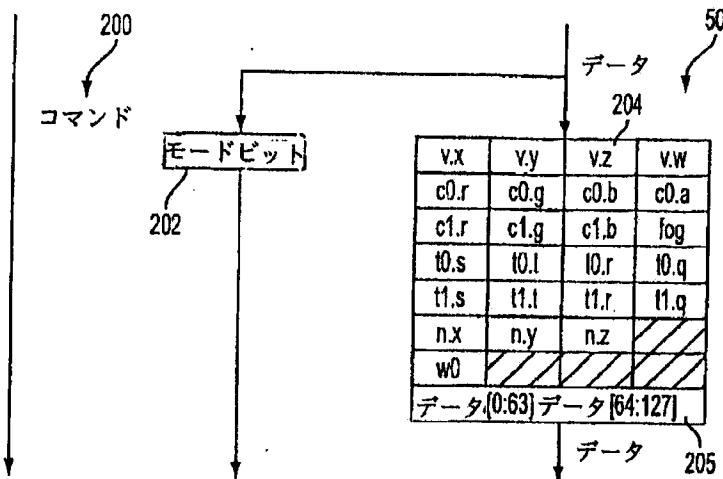
後

(従来技術)

【図1B】



【図2】

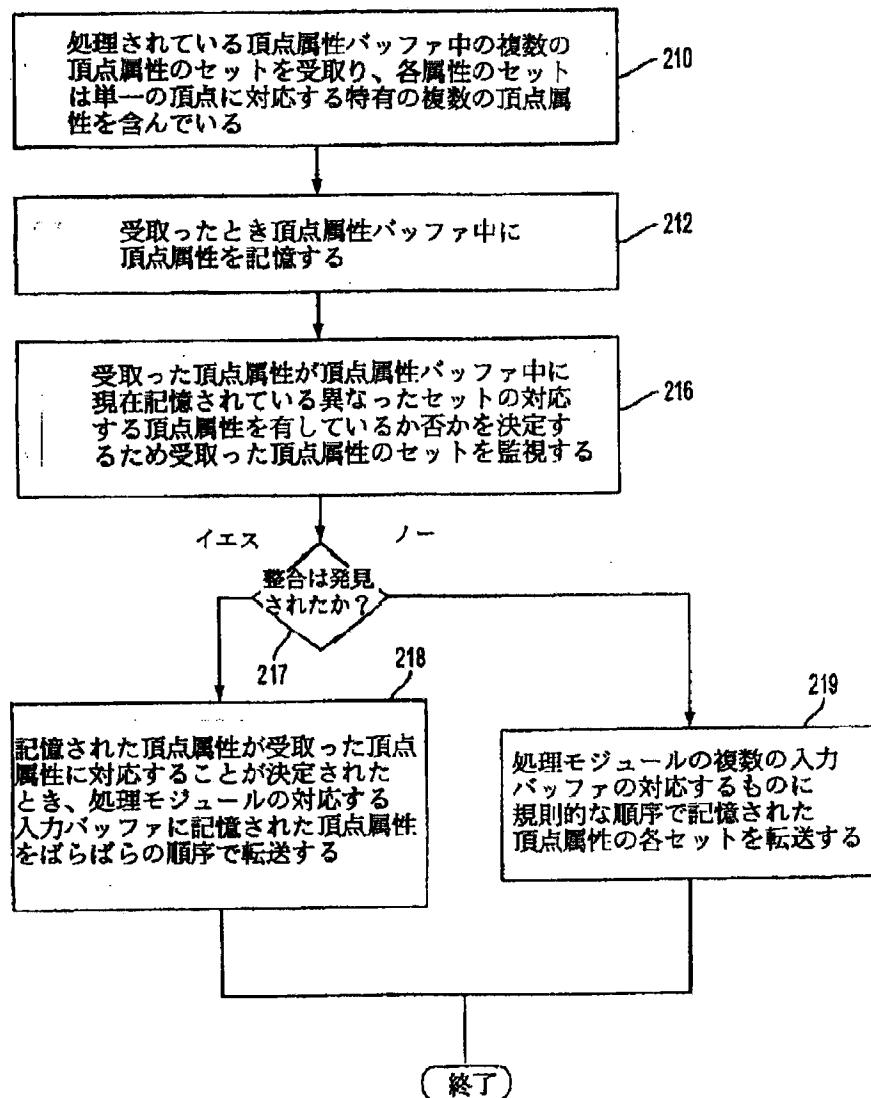


【図2A】

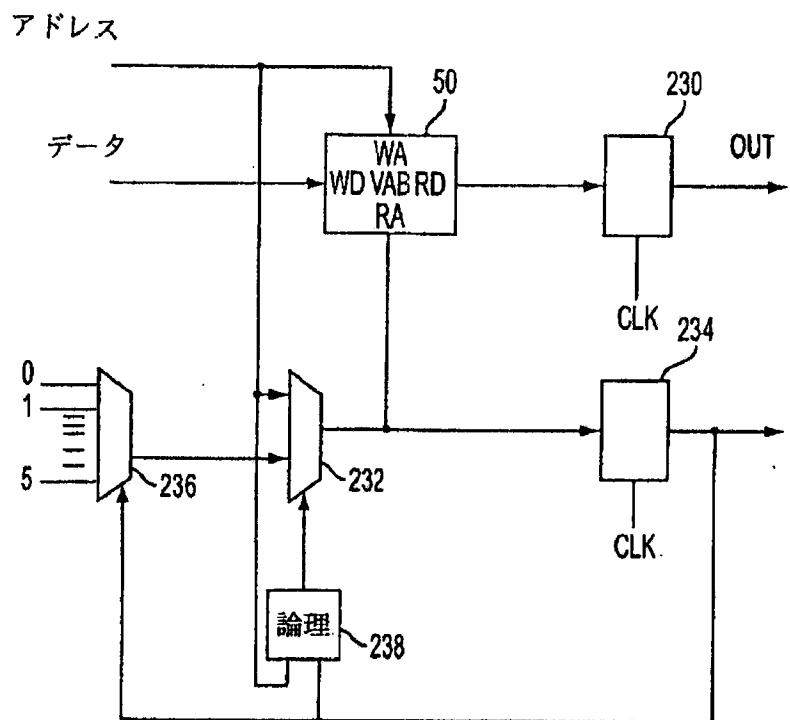
コマンド	変換	ライティング	記述
機能停止	機能停止		

FE2XF_CMD_NOP			NO OPERATION. CAN BE USED AS A SPACER BETWEEN COMMANDS.
FE2XF_CMD_VERTEX	READ	READ	VERTEX DATA.
FE2XF_CMD_PASSTHR			PASS THROUGH, TRANSFORM AND LIGHTING PASS THE DATA THROUGH.
FE2XF_CMD_RDVAB			READ THE VAB CONTENTS WHEN CONTEXT SWITCHING.
FE2XF_CMD_LDMODE			LOAD NEW MODE BITS.
FE2XF_CMD_LDXFCTX	WRITE		LOAD TRANSFORM CONTEXT MEMORY DATA.
FE2XF_CMD_RDXFCTX	READ		READ TRANSFORM CONTEXT MEMORY DATA.
FE2XF_CMD_LDLTCTX		WRITE	LOAD LIGHTING CONTEXT MEMORY DATA.
FE2XF_CMD_RDLTCTX		READ	READ LIGHTING CONTEXT MEMORY DATA.
FE2XF_CMD_LDLTC0		WRITE	LOAD LIGHTING CONTEXT0 MEMORY DATA.
FE2XF_CMD_RDLTC0		READ	READ LIGHTING CONTEXT0 MEMORY DATA.
FE2XF_CMD_LDLTC1		WRITE	LOAD LIGHTING CONTEXT1 MEMORY DATA.
FE2XF_CMD_RDLTC1		READ	READ LIGHTING CONTEXT1 MEMORY DATA.
FE2XF_CMD_LDLTC2		WRITE	LOAD LIGHTING CONTEXT2 MEMORY DATA.
FE2XF_CMD_RDLTC2		READ	READ LIGHTING CONTEXT2 MEMORY DATA.
FE2XF_CMD_LTLTC3		WRITE	LOAD LIGHTING CONTEXT3 MEMORY DATA.
FE2XF_CMD_RDLTC3		READ	READ LIGHTING CONTEXT3 MEMORY DATA.
FE2XF_CMD_SYNC	READ+ WRITE	READ+ WRITE	SIMILAR TO NOP, BUT IS NOT ALLOWED TO BE PROCESSED IN PARALLEL.

【図2B】



【図2C】

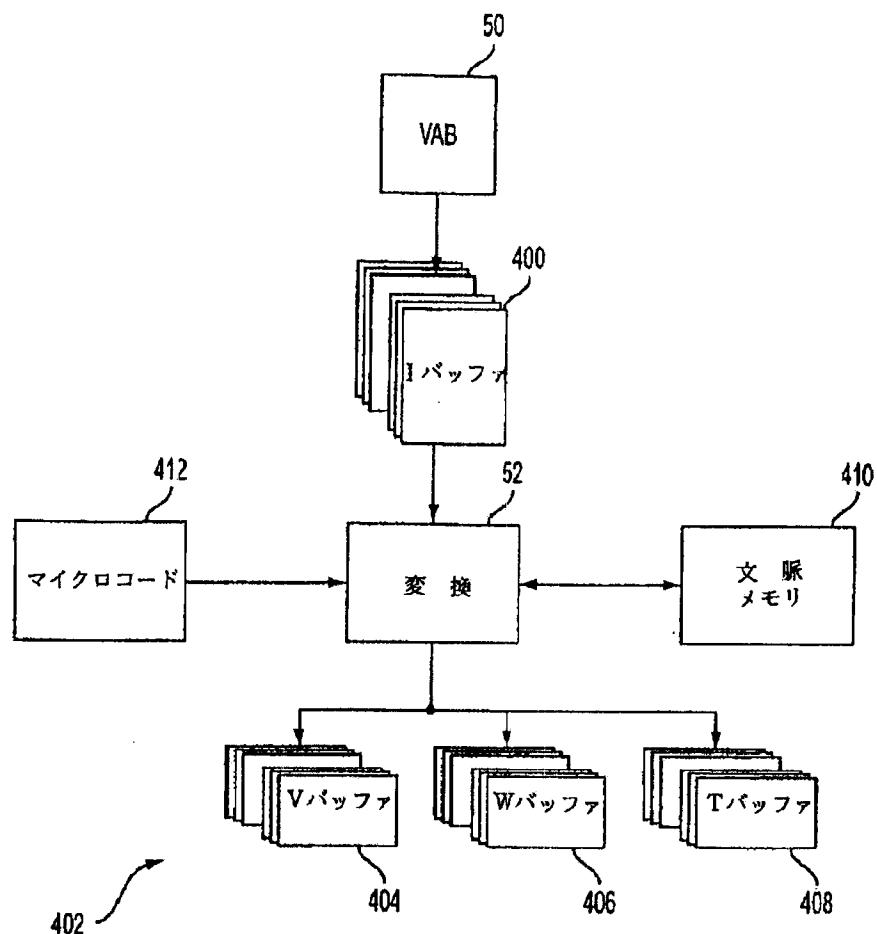


【図3】

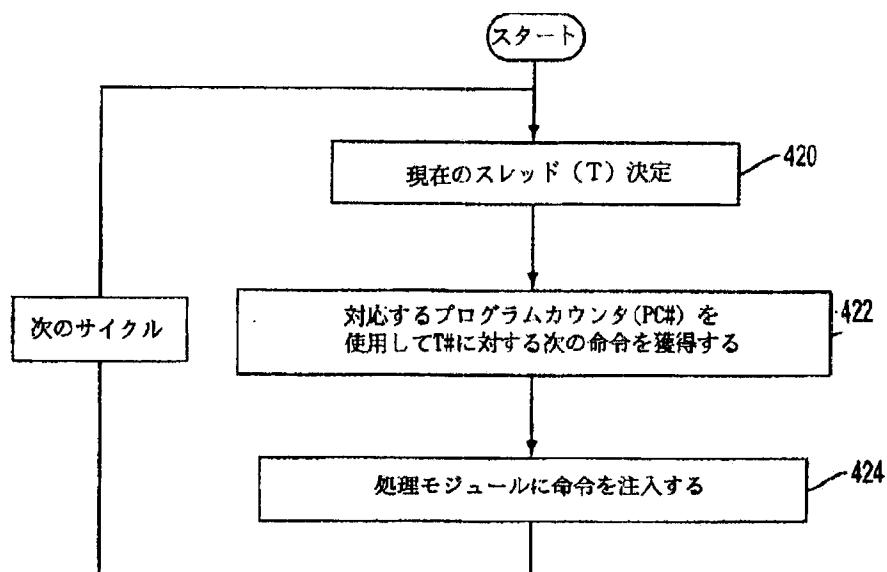
モードビット ビット 記述

T0	1	TEXTURE 0 ENABLE
TXF0	1	TEXTURE 0 MATRIX TRANSFORM ENABLE
TDV0	1	TEXTURE 0 w DIVIDE ENABLE
TOS	3	TEXTURE 0 TEXGENs CONTROL
TOT	3	TEXTURE 0 TEXGENt CONTROL
TOU	3	TEXTURE 0 TEXGENr CONTROL
TOQ	2	TEXTURE 0 TEXGENq CONTROL
T1	1	TEXTURE 1 ENABLE
TXF1	1	TEXTURE 1 MATRIX TRANSFORM ENABLE
TDV1	1	TEXTURE 1 w DIVIDE ENABLE
T1S	3	TEXTURE 1 TEXGENs CONTROL
T1T	3	TEXTURE 1 TEXGENt CONTROL
T1U	3	TEXTURE 1 TEXGENr CONTROL
T1Q	2	TEXTURE 1 TEXGENq CONTROL
ETY	1	EYE TYPE INFINITE(0) OR LOCAL(1)
LIT	1	LIGHTING ENABLE
NRM	1	NORMAL NORMALIZE ENABLE
FOG	1	FOG ENABLE
LIS	16	LIGHT STATUS (8 LIGHTS BY 2 BITS EACH, 0:OFF,1:INFINITE,2:LOCAL,3:SPOTLIGHT)
FG	2	FOGGEN CONTROL(0: OFF, 1:RADIAL, 2:PLANE)
LAT	1	LIGHT ATTENUATION CONTROL (0: INVERT, 1: NO INVERT)
C1I	1	SPECULAR COLOR INPUT ENABLE
C1O	1	SPECULAR COLOR OUTPUT ENABLE
CM	4	COLOR MATERIAL CONTROL (1: EMISSIVE, 2: AMBIENT, 4: DIFFUSE, 8:SPECULAR)
PP	1	POINT PARAMETER ENABLE
SKIN	1	SKINNING ENABLE
VPAS	1	VERTEX PASS ENABLE

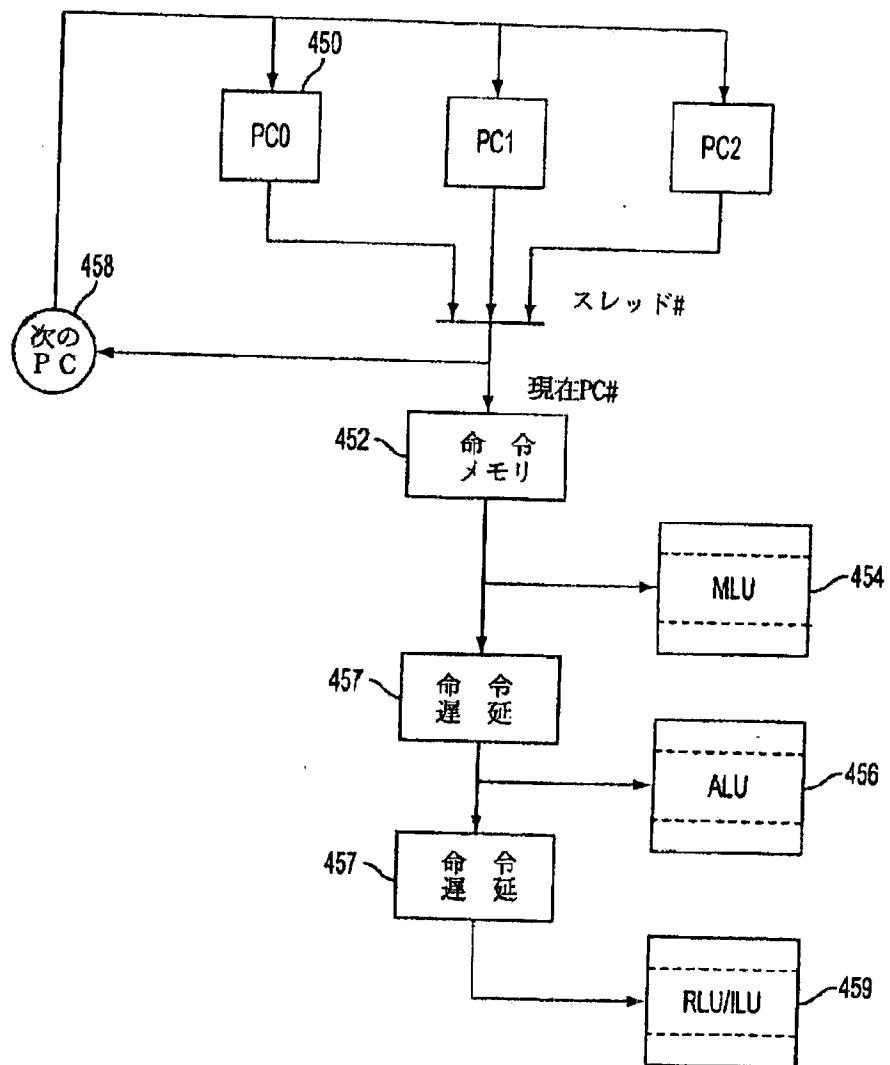
【図4】



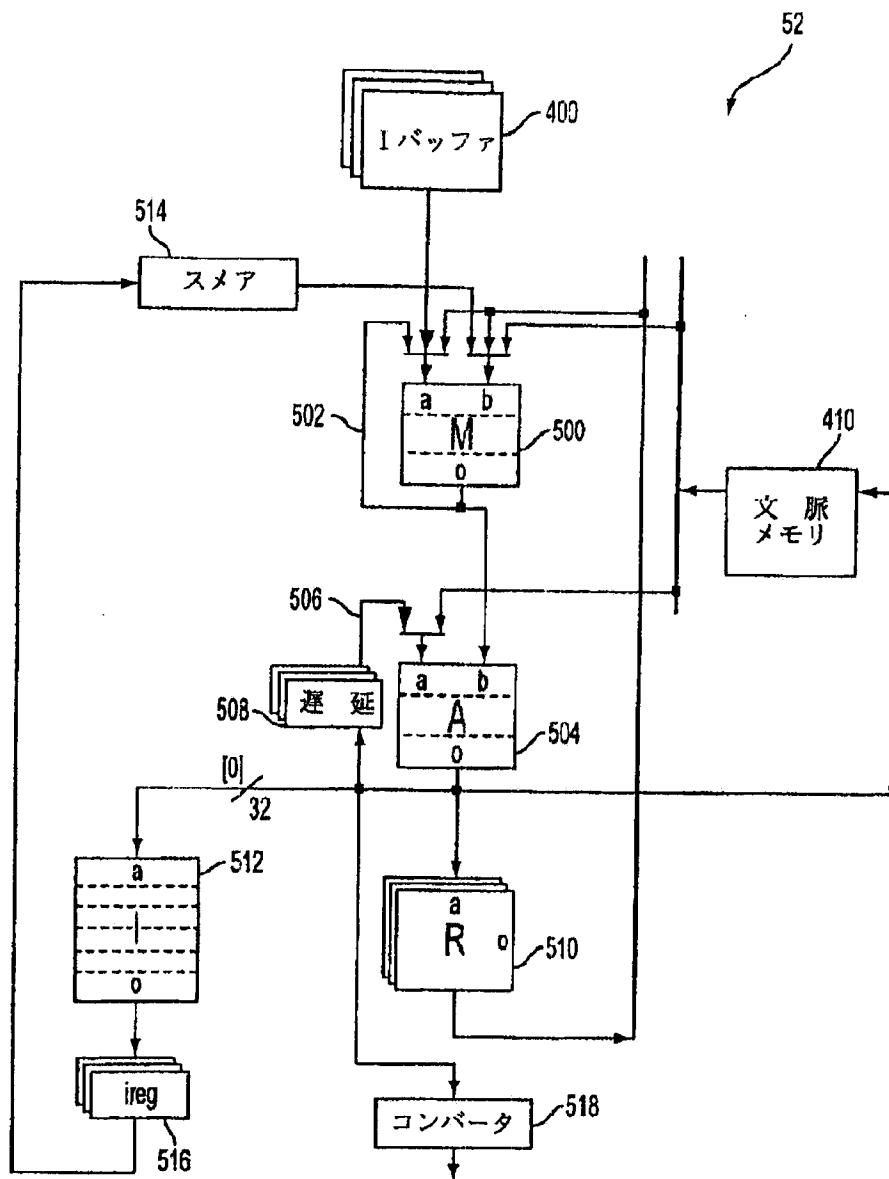
【図4 A】



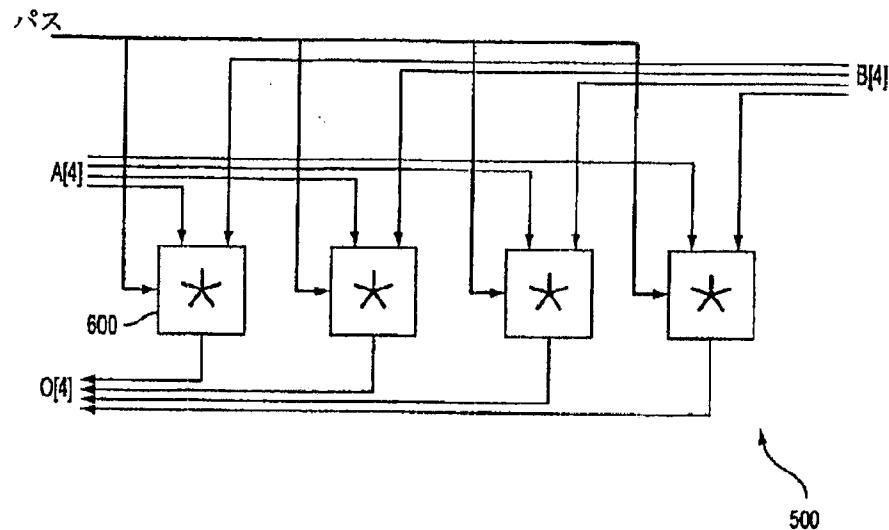
【图 4 B】



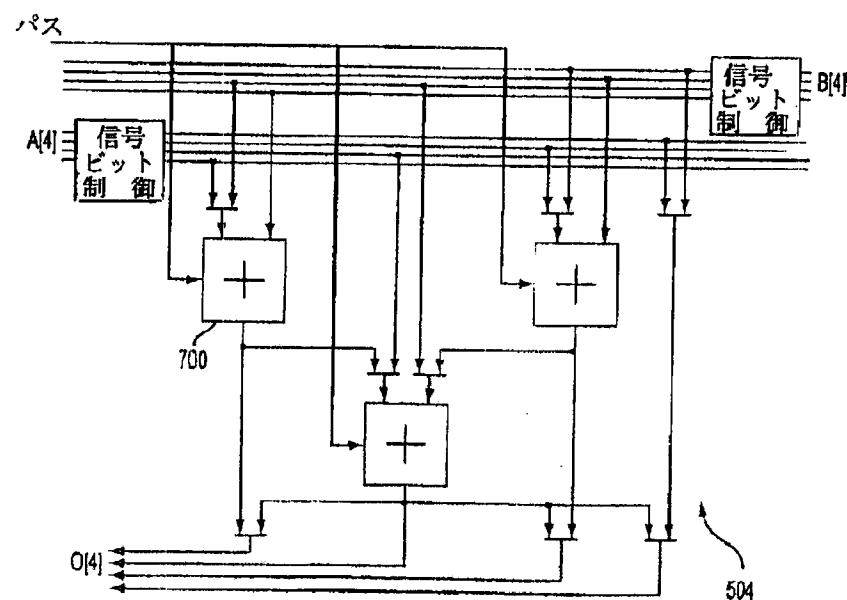
【図5】



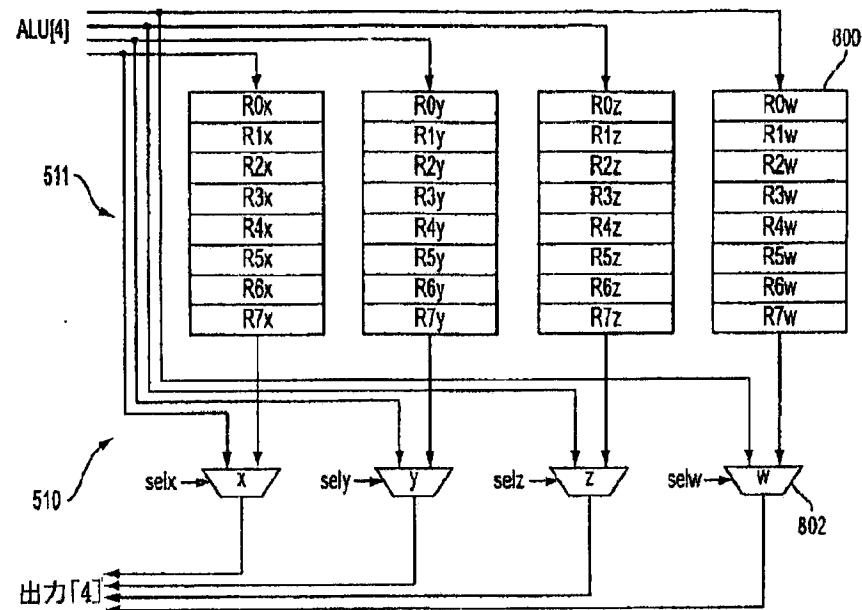
【図6】



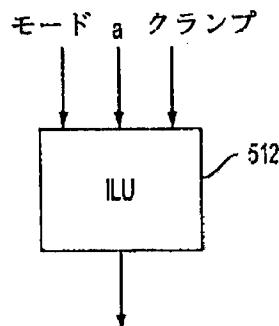
【図7】



【図8】



【図9】



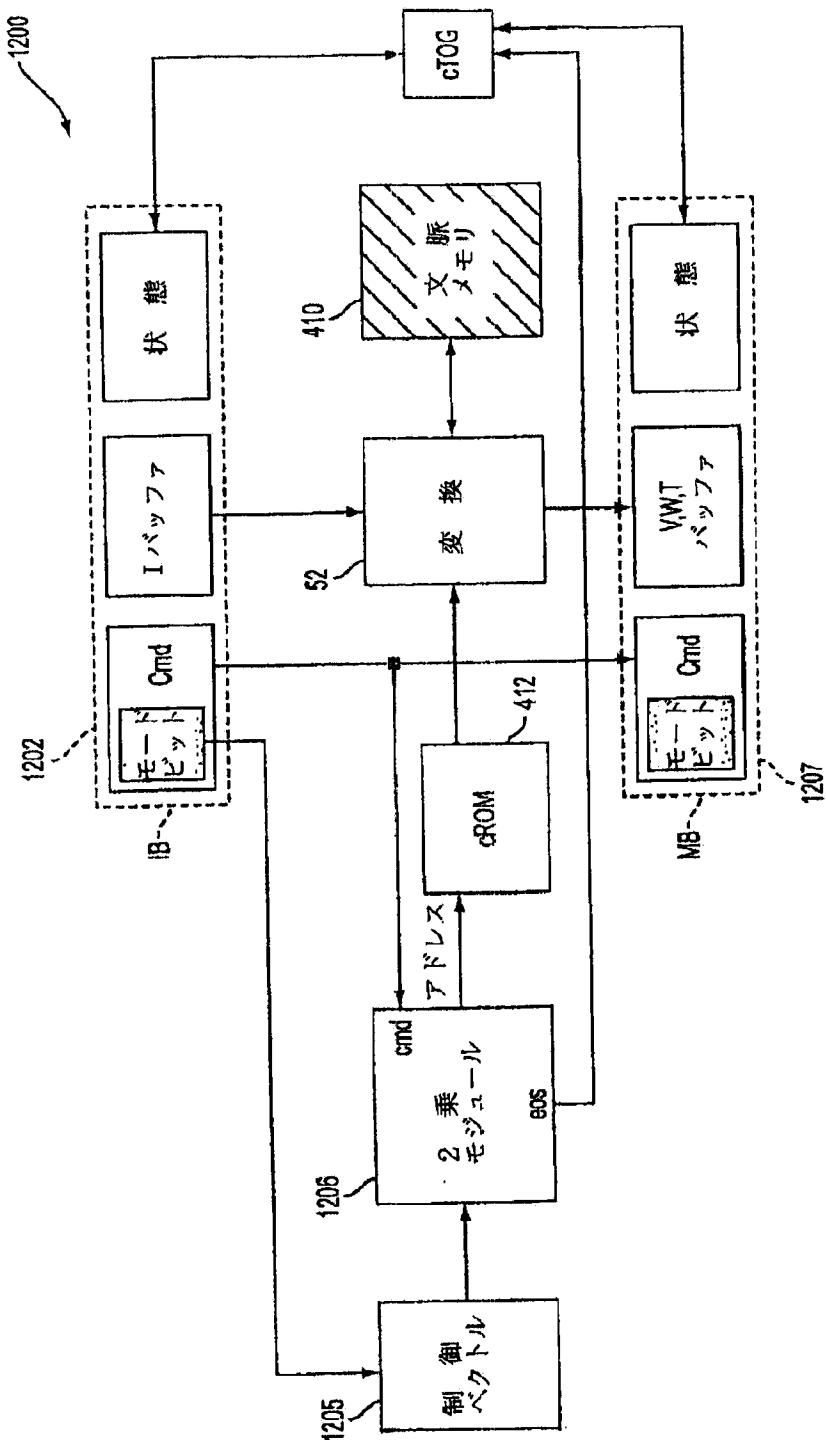
【図10】

アドレス	ターゲット	作用	記述
TPOS	TBUFFER	T[0] = ALU	POSITION
TT0	TBUFFER	T[3] = ALU	TEXTURE0
TT1	TBUFFER	T[4] = ALU	TEXTURE1
WEV	WBUFFER,VBUFFER	W[0] = ALU, V[0].y = ALU.w	EYE VECTOR
WLV0	WBUFFER,VBUFFER	W[1] = ALU, V[1].y = ALU.w	LIGHT0 DIRECTION VECTOR
WLV1	WBUFFER,VBUFFER	W[2] = ALU, V[2].y = ALU.w	LIGHT1 DIRECTION VECTOR
WLV2	WBUFFER,VBUFFER	W[3] = ALU, V[3].y = ALU.w	LIGHT2 DIRECTION VECTOR
WLV3	WBUFFER,VBUFFER	W[4] = ALU, V[4].y = ALU.w	LIGHT3 DIRECTION VECTOR
WLV4	WBUFFER,VBUFFER	W[5] = ALU, V[5].y = ALU.w	LIGHT4 DIRECTION VECTOR
WLV5	WBUFFER,VBUFFER	W[6] = ALU, V[6].y = ALU.w	LIGHT5 DIRECTION VECTOR
WLV6	WBUFFER,VBUFFER	W[7] = ALU, V[7].y = ALU.w	LIGHT6 DIRECTION VECTOR
WLV7	WBUFFER,VBUFFER	W[8] = ALU, V[8].y = ALU.w	LIGHT7 DIRECTION VECTOR
WSL0	WBUFFER	W[9] = ALU	SPOTLIGHT0 cos
WSL1	WBUFFER	W[10] = ALU	SPOTLIGHT1 cos
WSL2	WBUFFER	W[11] = ALU	SPOTLIGHT2 cos
WSL3	WBUFFER	W[12] = ALU	SPOTLIGHT3 cos
WSL4	WBUFFER	W[13] = ALU	SPOTLIGHT4 cos
WSL5	WBUFFER	W[14] = ALU	SPOTLIGHT5 cos
WSL6	WBUFFER	W[15] = ALU	SPOTLIGHT6 cos
WSL7	WBUFFER	W[16] = ALU	SPOTLIGHT7 cos
VED	VBUFFER	V[0].x = 1.0, V[0].z = ALU.w	EYE RADIAL DISTANCE VECTOR
VLD0	VBUFFER	V[1].x = 1.0, V[1].z = ALU.w	LIGHT0 DISTANCE VECTOR
VLD1	VBUFFER	V[2].x = 1.0, V[2].z = ALU.w	LIGHT1 DISTANCE VECTOR
VLD2	VBUFFER	V[3].x = 1.0, V[3].z = ALU.w	LIGHT2 DISTANCE VECTOR
VLD3	VBUFFER	V[4].x = 1.0, V[4].z = ALU.w	LIGHT3 DISTANCE VECTOR
VLD4	VBUFFER	V[5].x = 1.0, V[5].z = ALU.w	LIGHT4 DISTANCE VECTOR
VLD5	VBUFFER	V[6].x = 1.0, V[6].z = ALU.w	LIGHT5 DISTANCE VECTOR
VLD6	VBUFFER	V[7].x = 1.0, V[7].z = ALU.w	LIGHT6 DISTANCE VECTOR
VLD7	VBUFFER	V[8].x = 1.0, V[8].z = ALU.w	LIGHT7 DISTANCE VECTOR
VC0	VBUFFER,TBUFFER	V[9] = ALU, T[1] = ALU	DIFFUSE COLOR
VC1	VBUFFER,TBUFFER	V[10] = ALU, T[2] = ALU	SPECULAR COLOR
VNRM	VBUFFER	V[11] = ALU	NORMAL VECTOR
VED2	VBUFFER	V[12] = ALU	EYE PLANAR DISTANCE VECTOR
TVW_NOP			NO VALID OUTPUT.

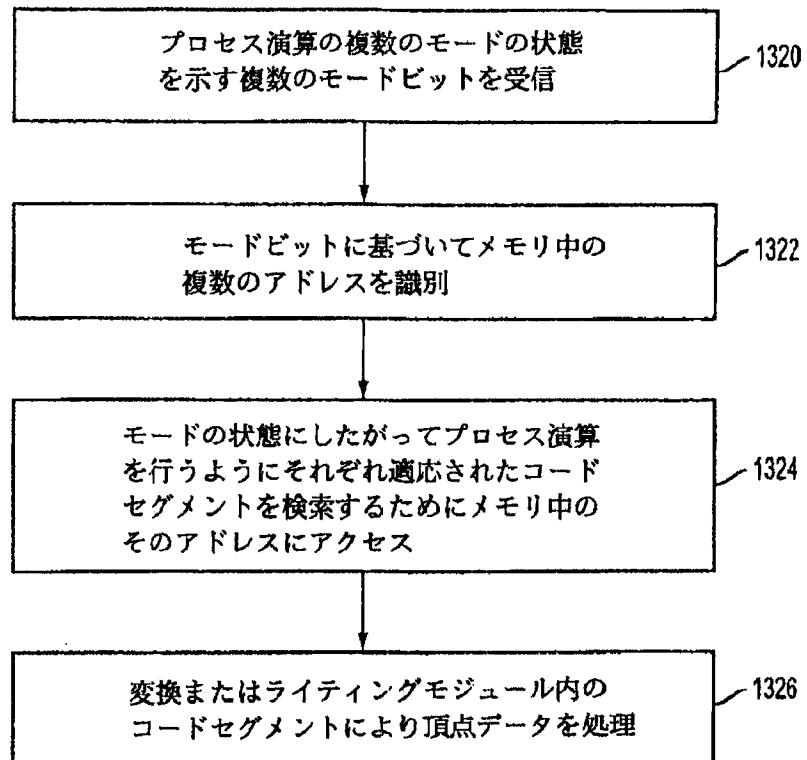
【図11】

マイクロコード フィールド	ビット位 置	遷延	記述
oa	6	0:5	2 OUTPUT BUFFER WRITE ADDRESS
rra	3	6:8	0 RLU READ ADDRESS
rwm	4	9:12	2 RLU WRITE MASK
rwa	3	13:15	2 RLU WRITE ADDRESS
ilu	2	16:17	2 ILU OPERATION
alu	4	18:21	1 ALU OPERATION
ais	2	22:23	1 ALU SIGN CONTROL
aia	1	24	1 ALU INPUT A MUX
mlu	3	25:27	0 MLU OPERATION
mib	2	28:29	0 MLU INPUT B MUX
mia	2	30:31	0 MLU INPUT A MUX
va	3	32:34	0 INPUT BUFFER READ ADDRESS
ce	1	35	0,2 CONTEXT MEMORY READ/WRITE
ca	6	36:41	0,2 CONTEXT MEMORY ADDRESS
mr	2	42:43	0 MLU INPUT VECTOR ROTATE

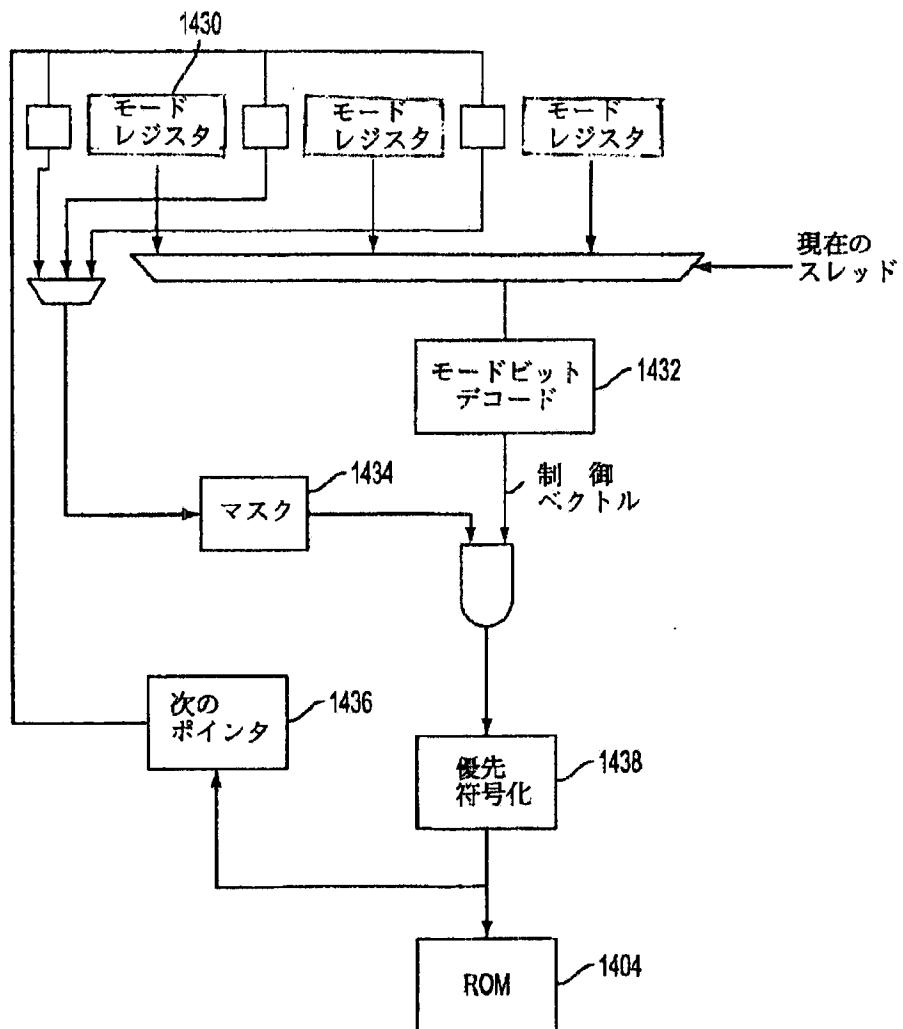
【図12】



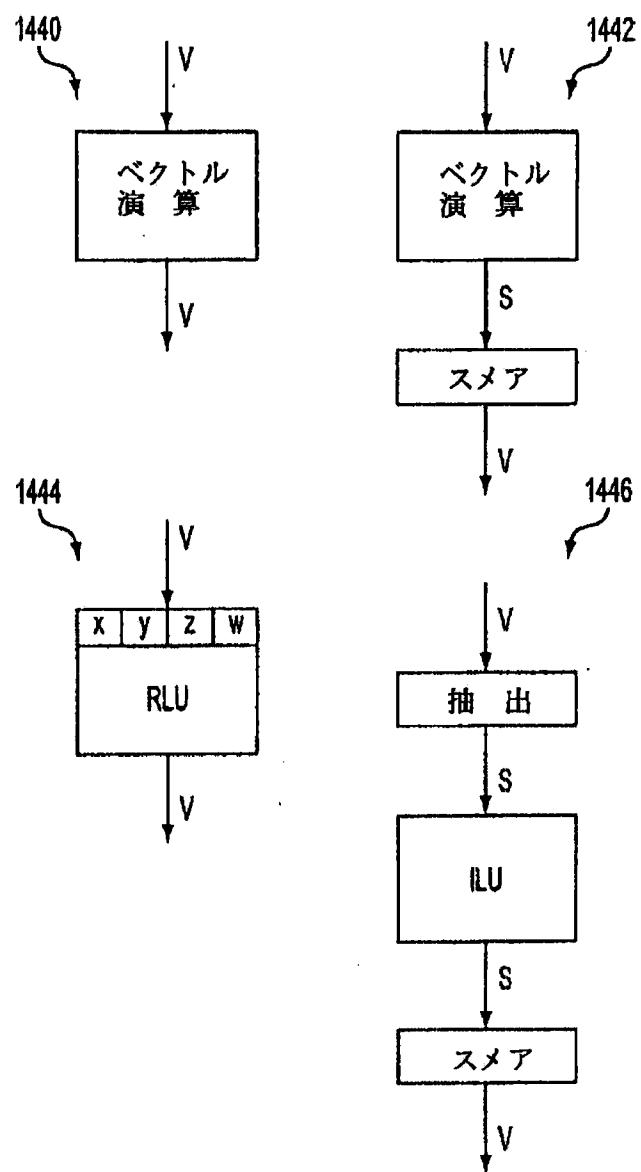
【図13】



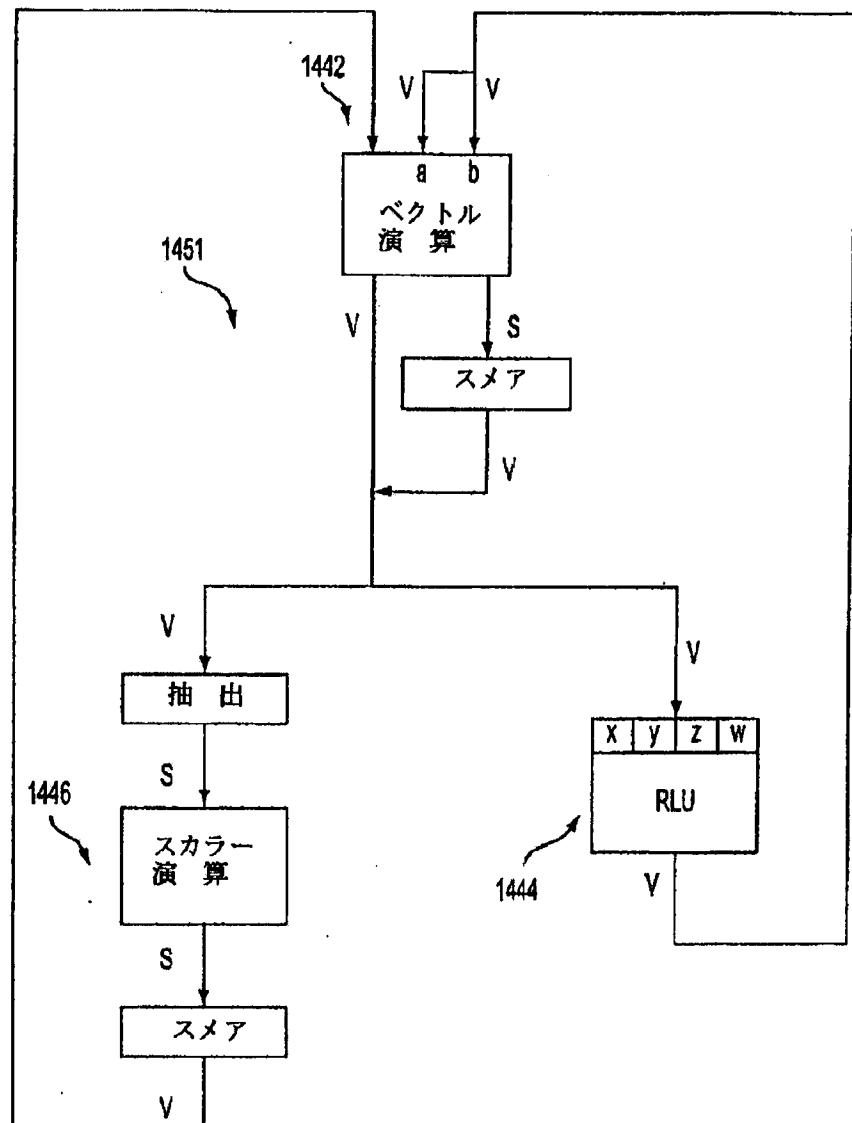
【図14】



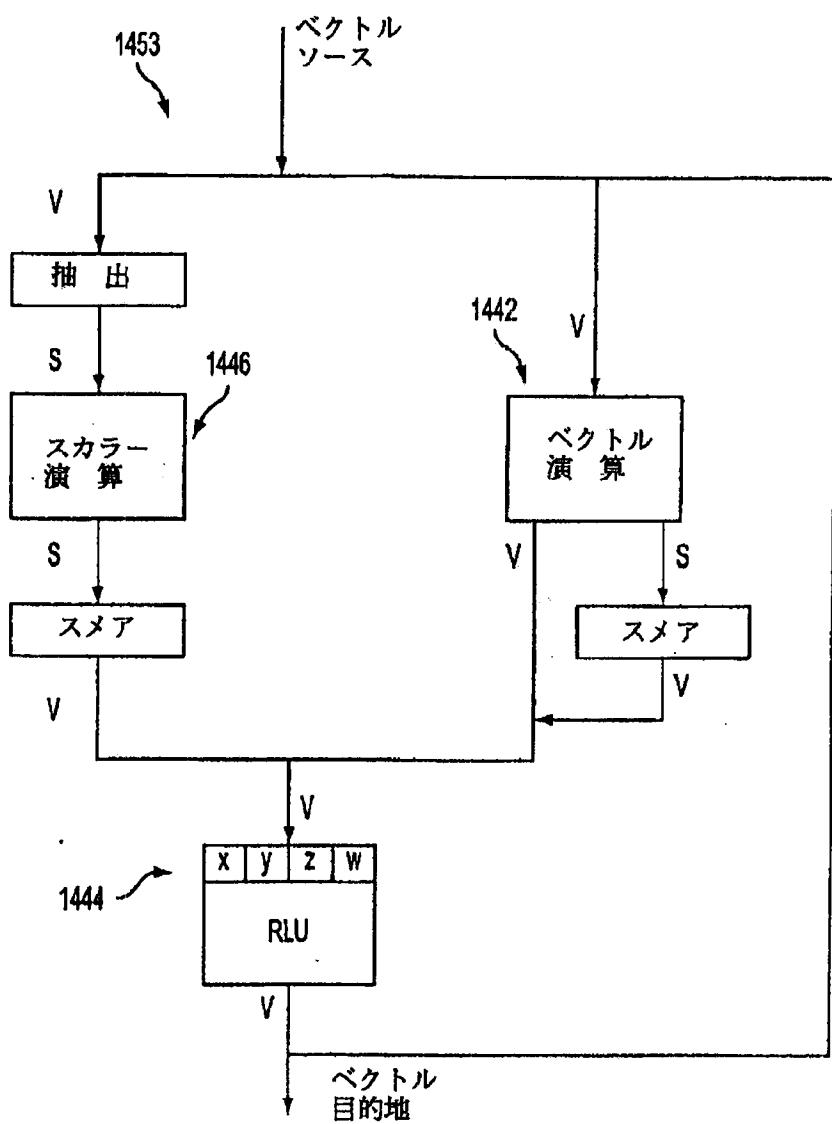
【図14A】



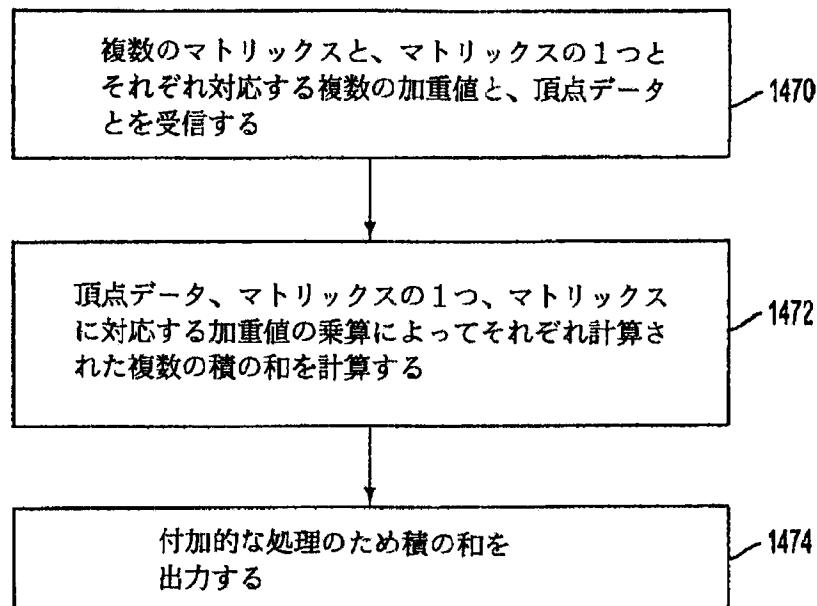
【図14B】



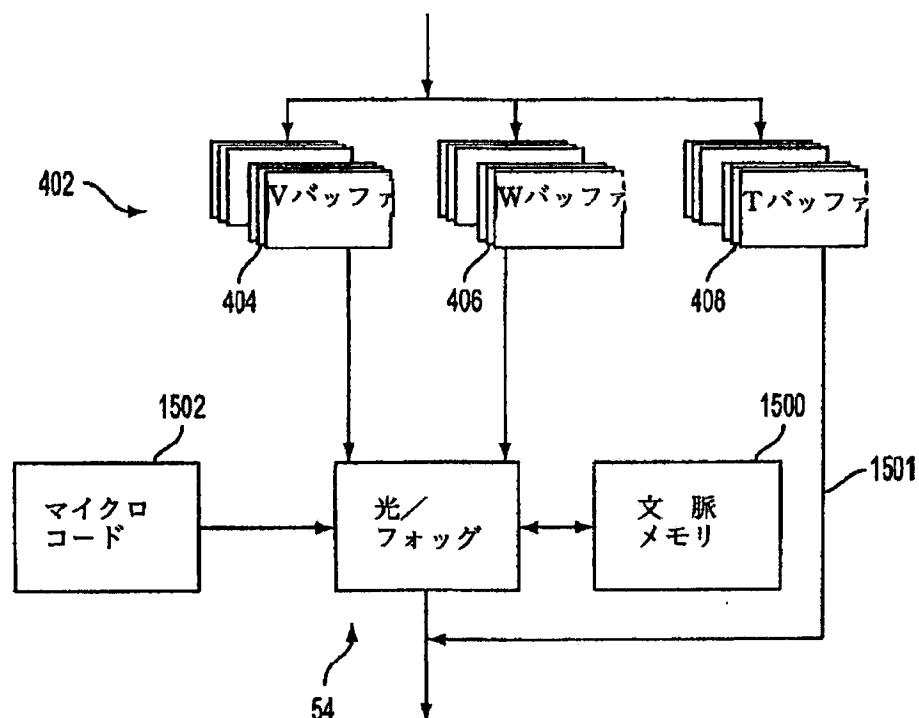
【四】14C】



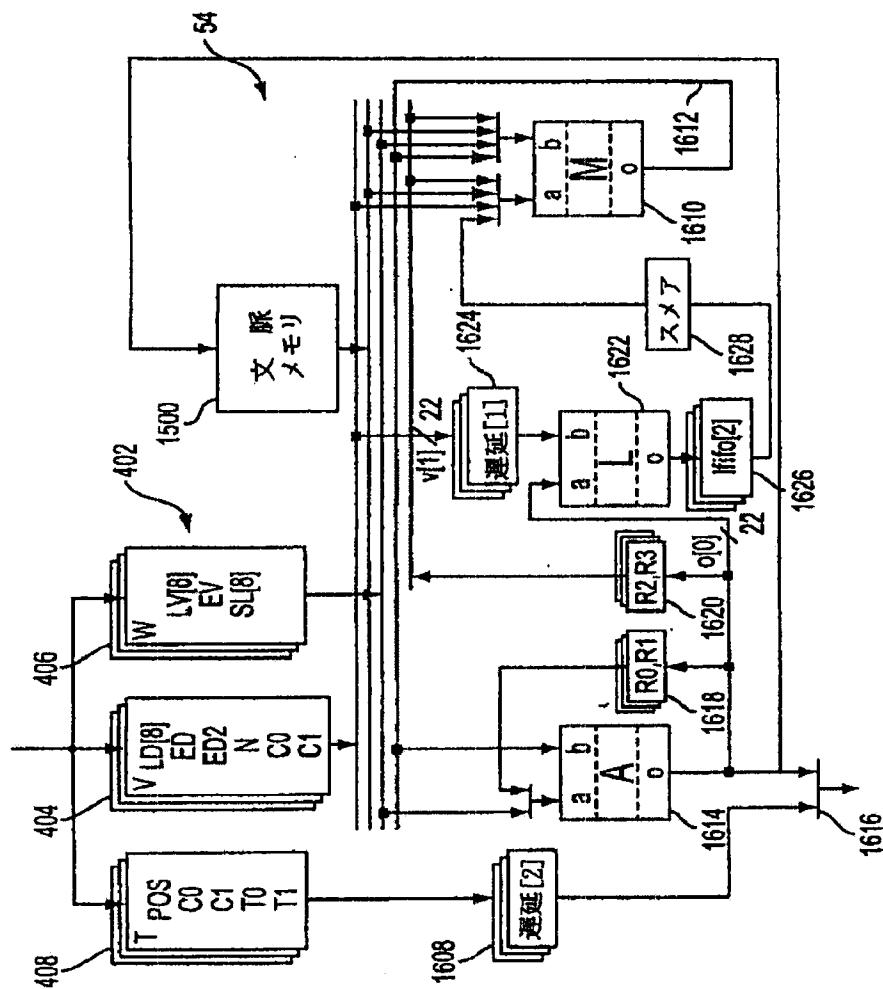
【図14D】



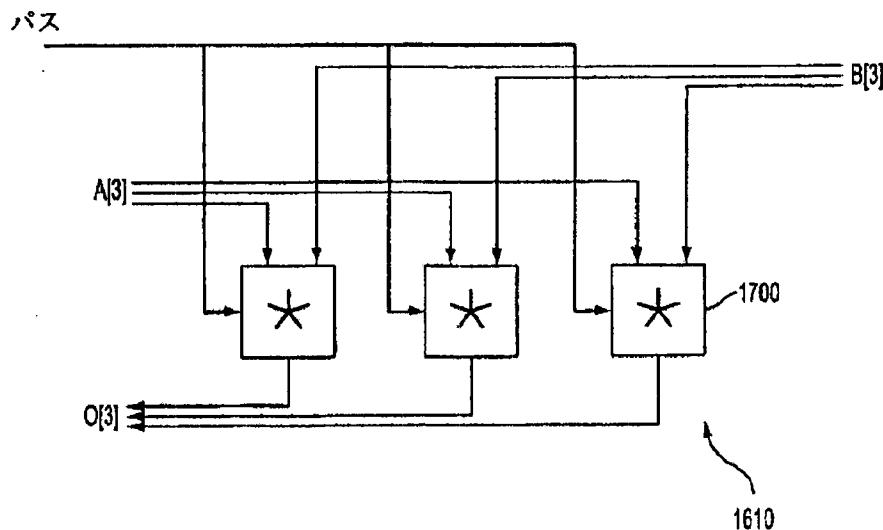
【図15】



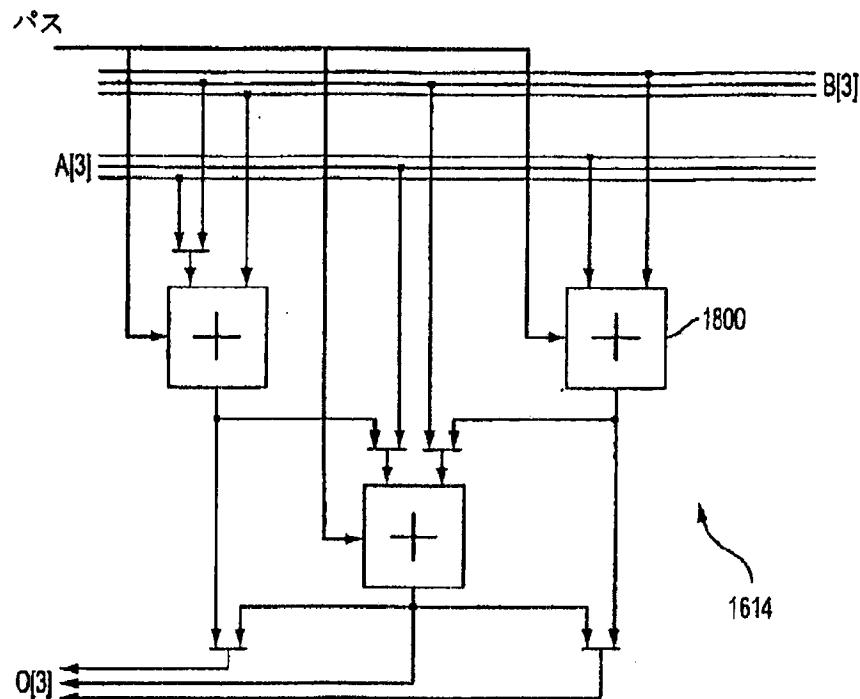
【四】 1 6】



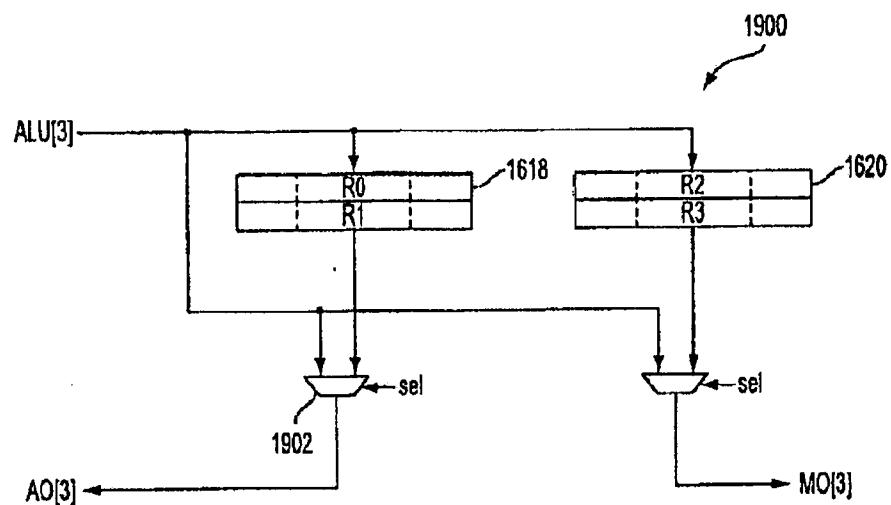
【図 17】



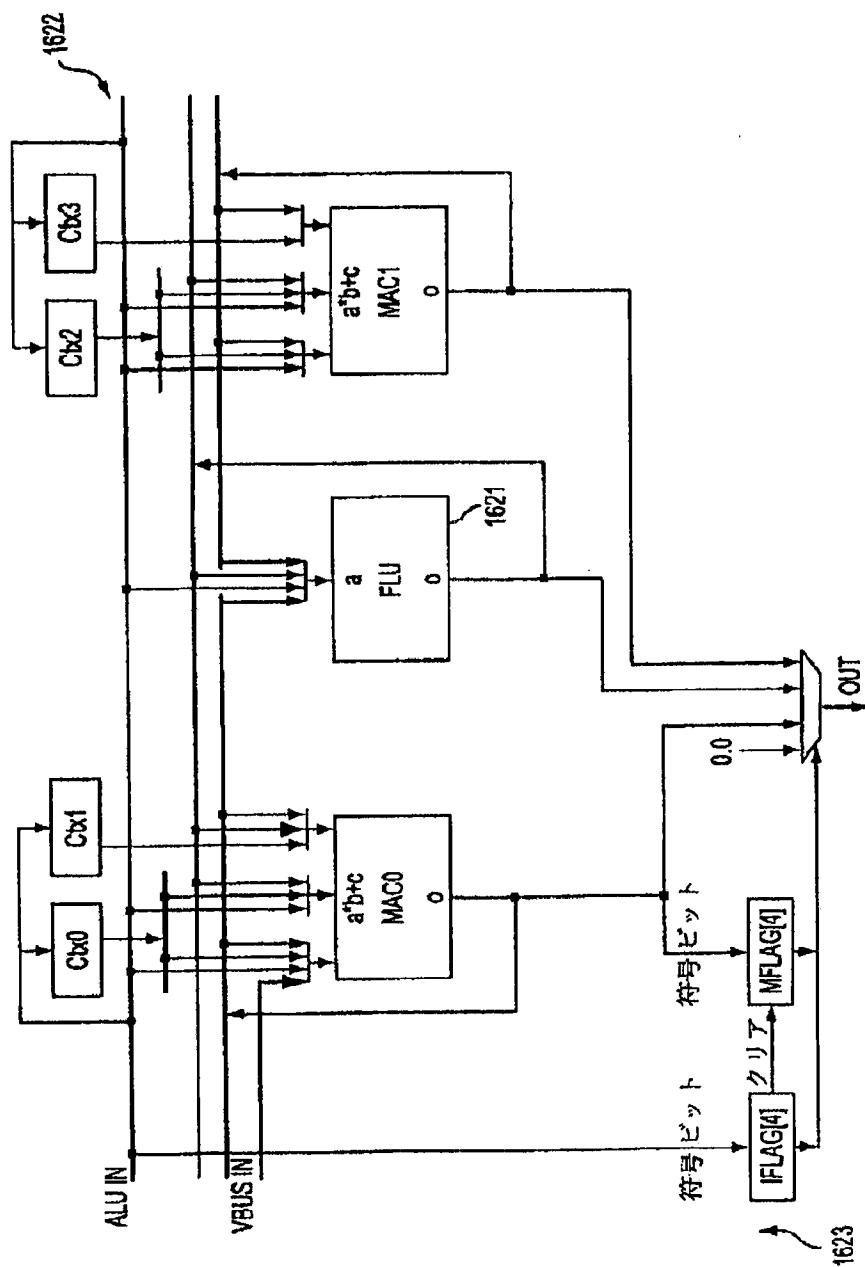
【図18】



【図19】



【図20】



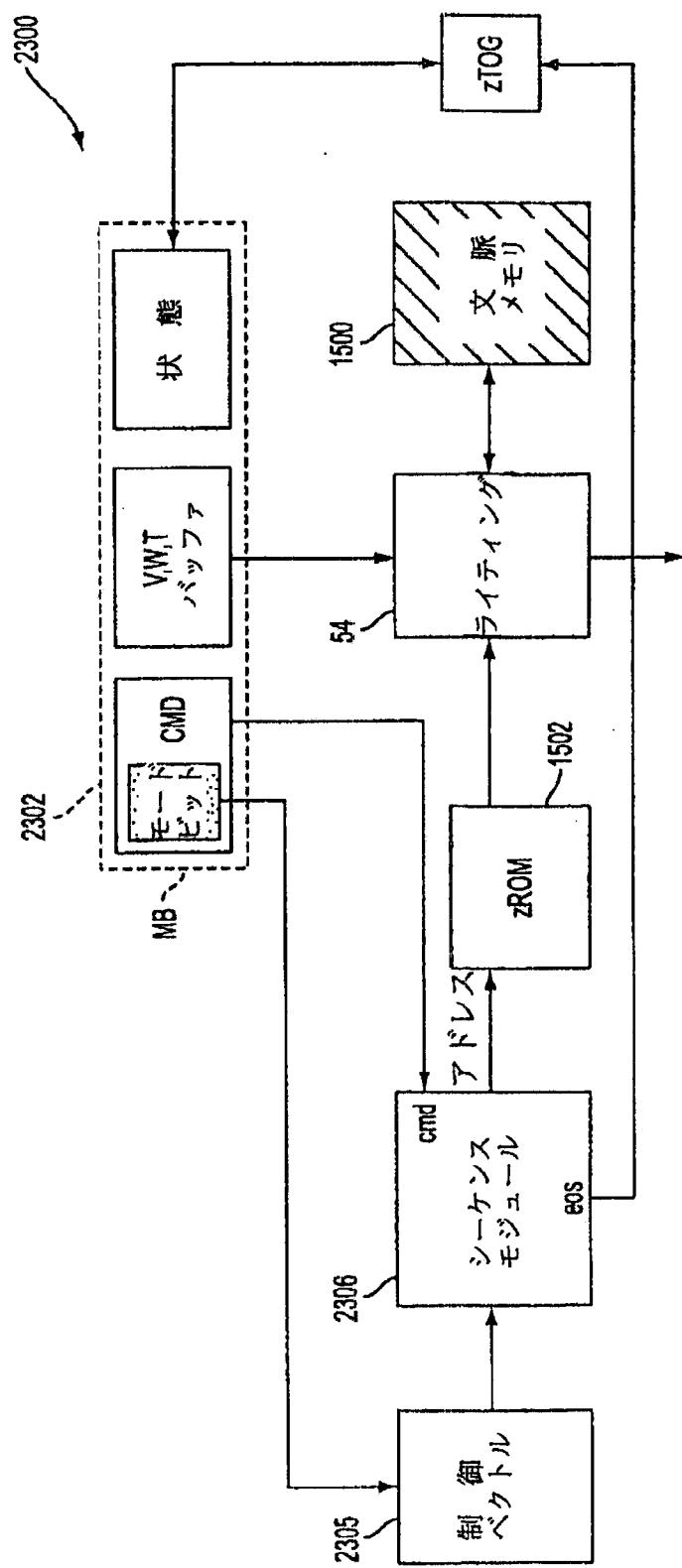
【図21】

名称	レジスタ	記述
Z	FLAG	CLEAR FLAG. SETS IF FLAG AND MFLAG TO 0.
C	FLAG	SPOTLIGHT CONE FLAG. SET IF VERTEX IS OUTSIDE SPOTLIGHT CONE.
S	FLAG	SPECULAR2 FLAG. SET IF SPECULAR CONTRIBUTION IS NEGATIVE.
D	FLAG	DIFFUSE FLAG. SET IF DIFFUSE TERM IS NEGATIVE.
	MFLAG	
U	MFLAG	SPOTLIGHT CONE ATTENUATION FLAG. SET IF SPOTLIGHT CONE ATTENUATION CONTRIBUTION IS NEGATIVE.
T	MFLAG	SPECULAR FLAG. SET IF SPECULAR CONTRIBUTION IS NEGATIVE.
R	MFLAG	RANGE FLAG. SET IF VERTEX IS TOO FAR AWAY FROM THE LIGHT.

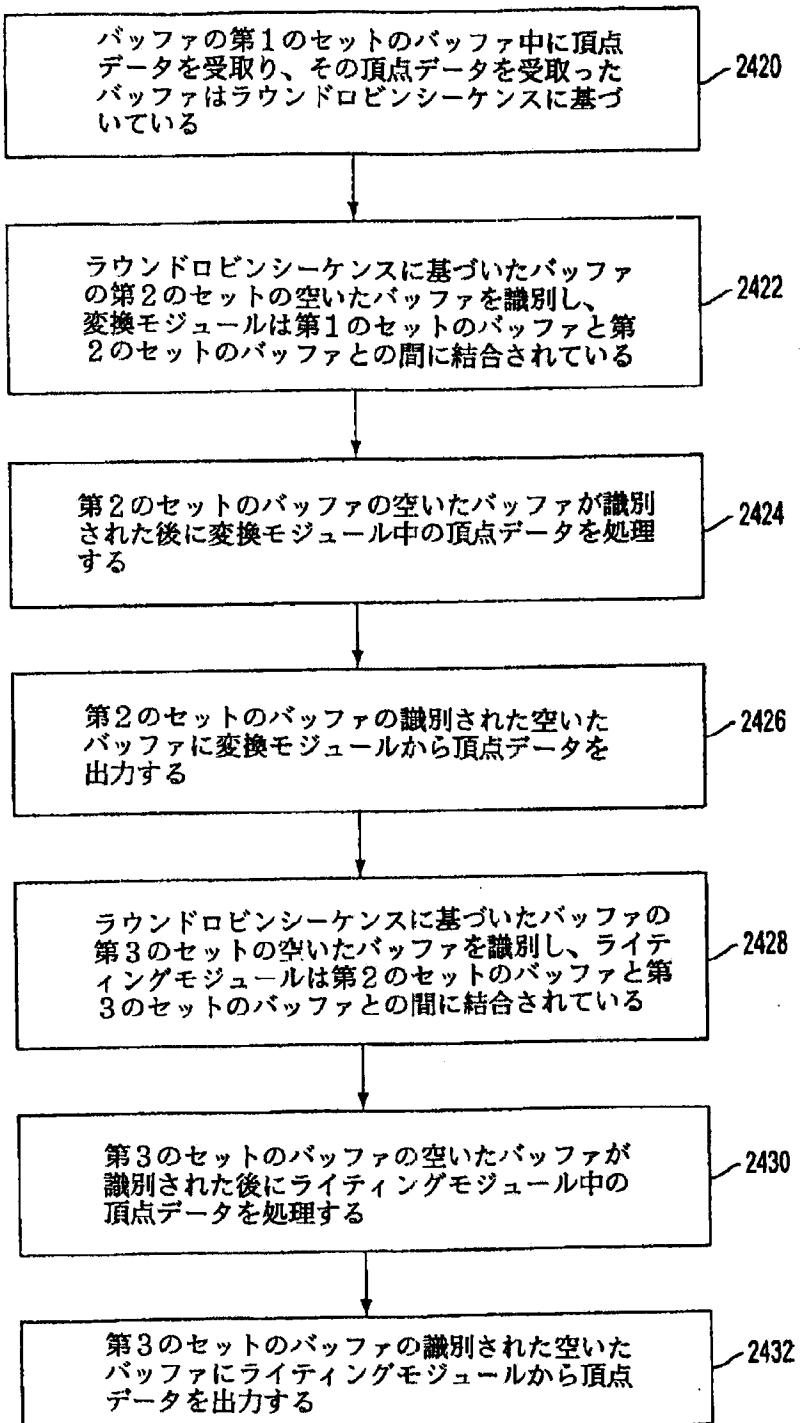
【図22】

マイクロコード フィールド	ビット	位置	遅延	記述
oa	3	0:2	2	OUTPUT ADDRESS
nwe	1	3	2	RLU WRITE ENABLE
nwa	2	4:5	2	RLU WRITE ADDRESS
R23	1	6	0	RLU (MLU) READ ADDRESS
R01	1	7	1	RLU (ALU) READ ADDRESS
aia	1	8	1	ALU INPUT A MUX
alu	2	9:10	1	ALU OPERATION
mib	2	11:12	0	MLU INPUT B MUX
mia	2	13:14	0	MLU INPUT A MUX
mlu	2	15:16	0	MLU OPERATION
mwa	5	17:21	0	MLU WBUFFER READ ADDRESS
awa	5	22:26	1	ALU WBUFFER READ ADDRESS
va	4	27:30	0	VBUFFER READ ADDRESS
os	2	31:32	2	LLU OUTPUT ADDRESS
frm	6	33:38	2	FLAG REGISTER MASK
mfe	1	39	2	MFLAG WRITE ENABLE
mfa	2	40:41	2	MFLAG WRITE ADDRESS
ife	1	42	2	IFLAG WRITE ENABLE
ifa	2	43:44	2	IFLAG WRITE ADDRESS
fia	2	45:46	2	FLU INPUT A MUX
flu	3	47:49	2	FLU OPERATION
M1c	1	50	2	MAC1 INPUT C MUX
M1b	2	51:52	2	MAC1 INPUT B MUX
M1a	2	53:54	2	MAC1 INPUT A MUX
M0c	2	55:56	2	MAC0 INPUT C MUX
M0b	2	57:58	2	MAC0 INPUT B MUX
M0a	2	59:60	2	MAC0 INPUT A MUX
ce	3	61:63	0,2	CONTEXT MEMORY READ/WRITE ENABLE
ca	6	64:69	0,2	CONTEXT MEMORY ADDRESS
C3a	4	70:73	2	CONTEXT3 MEMORY ADDRESS
C2a	4	74:77	2	CONTEXT2 MEMORY ADDRESS
C1a	5	78:82	2	CONTEXT1 MEMORY ADDRESS
C0a	2	83:84	2	CONTEXT0 MEMORY ADDRESS

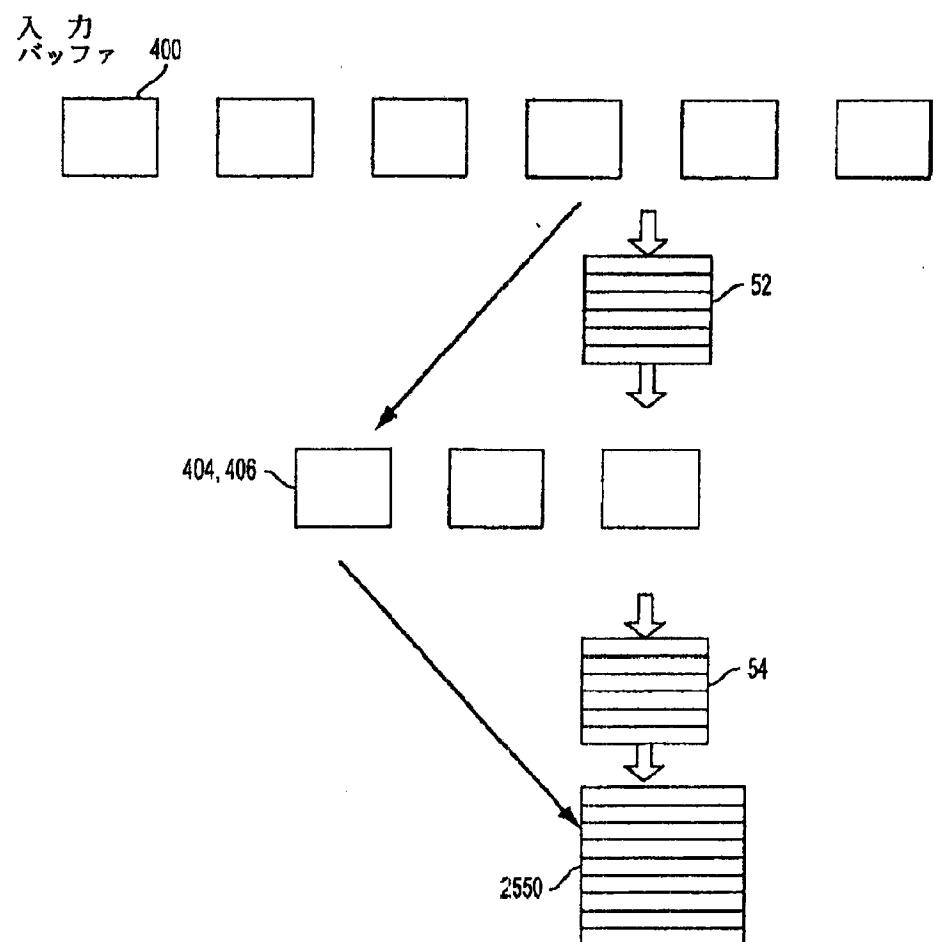
【図23】



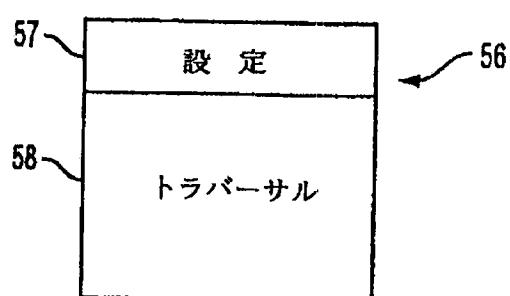
【図24】



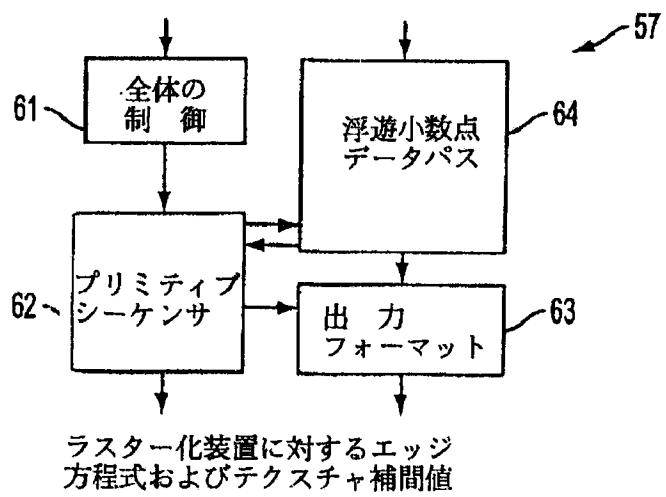
【図25】



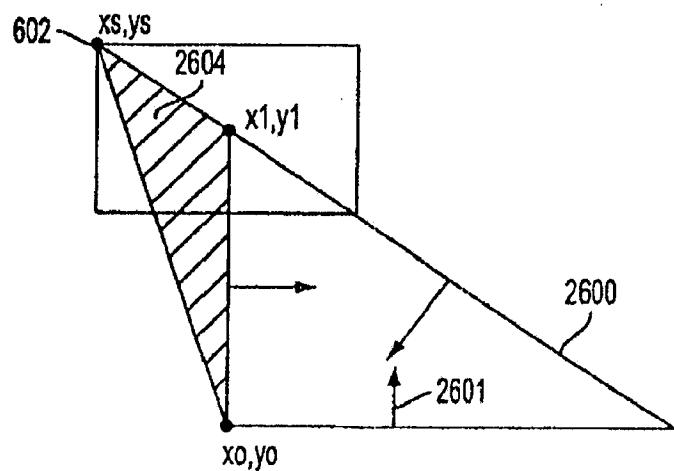
【図25B】



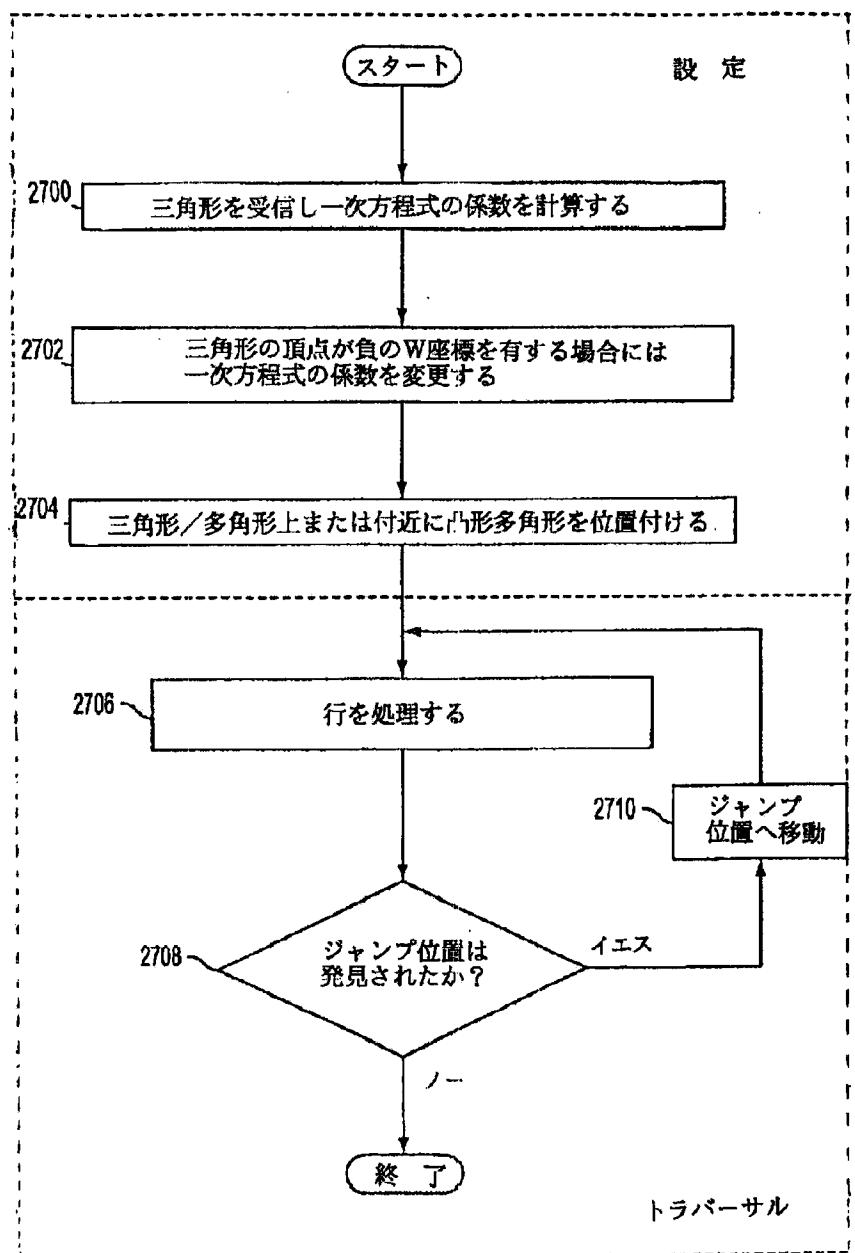
【図26】



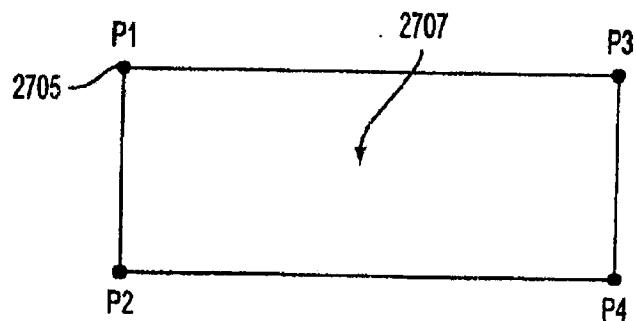
【図26A】



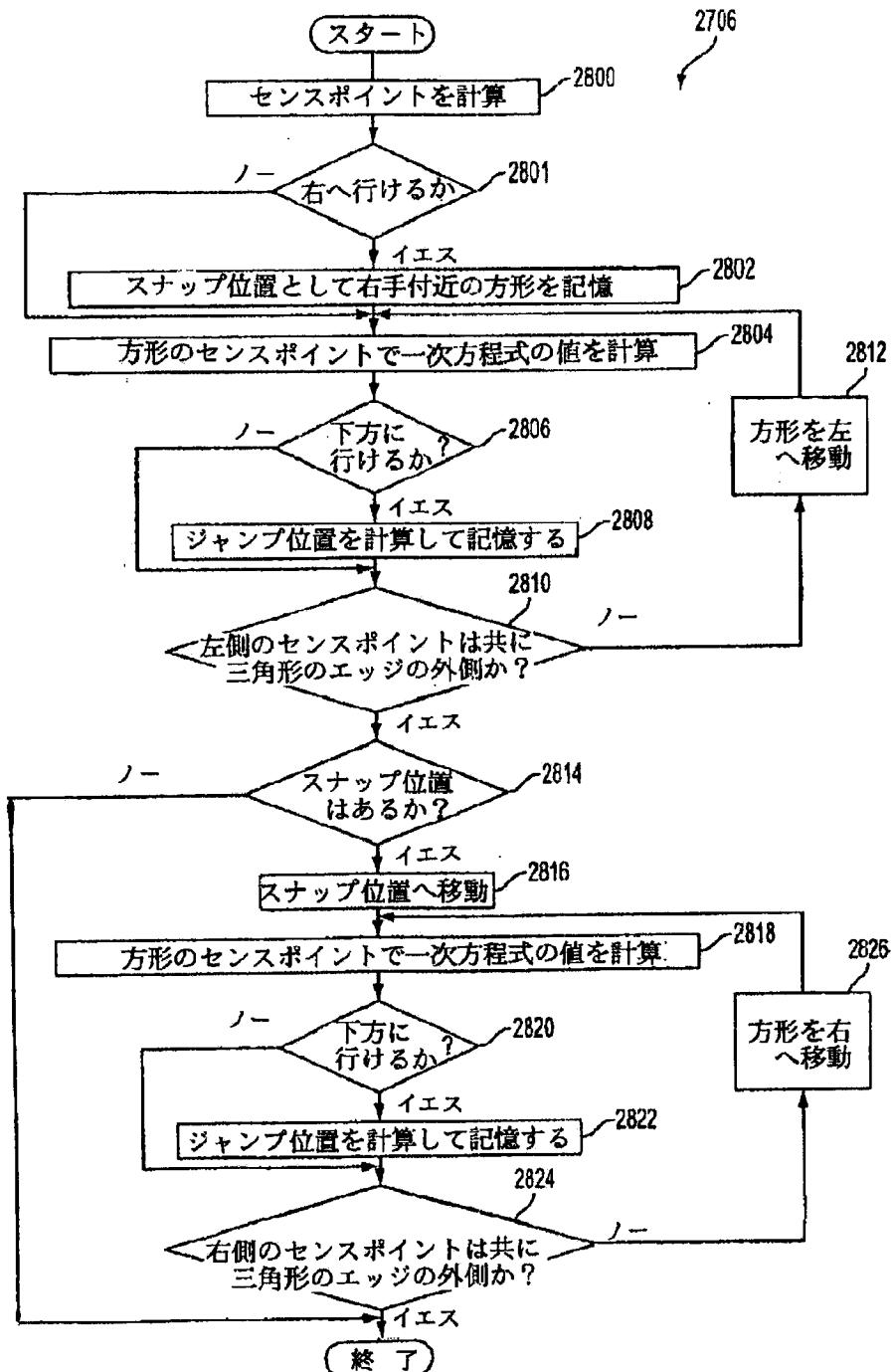
【図27】



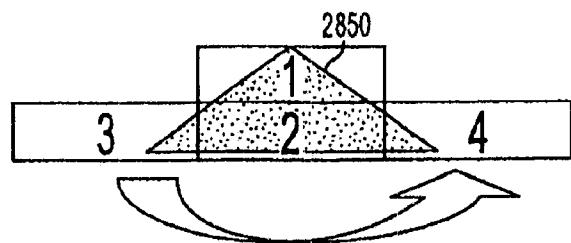
【図27A】



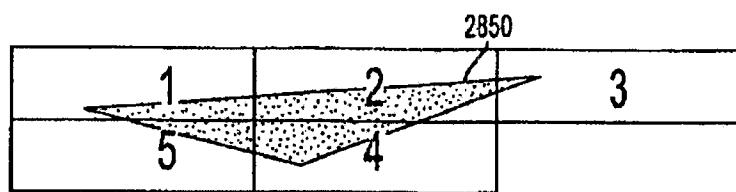
【 2 8】



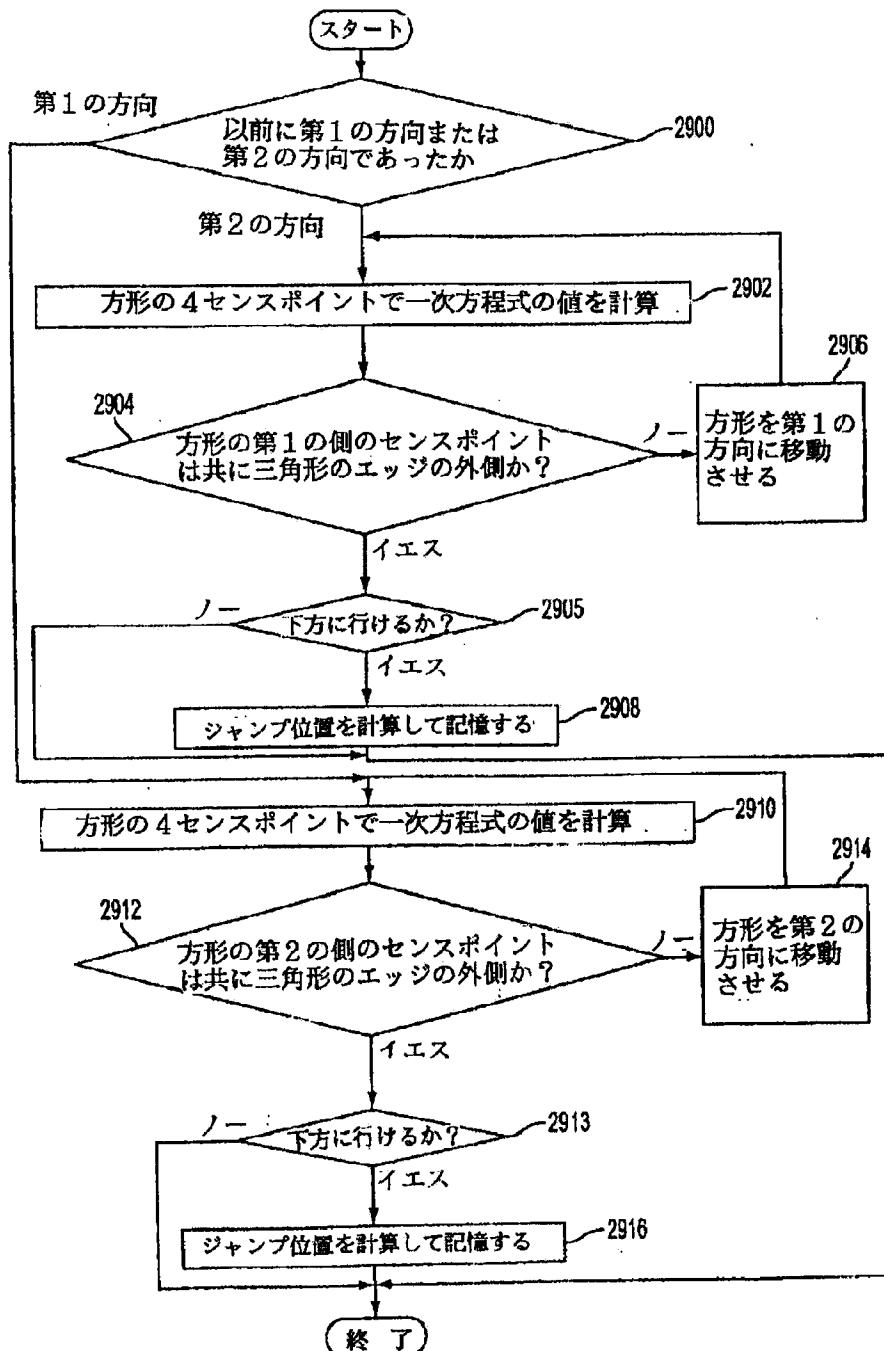
【図28A】



【図28B】



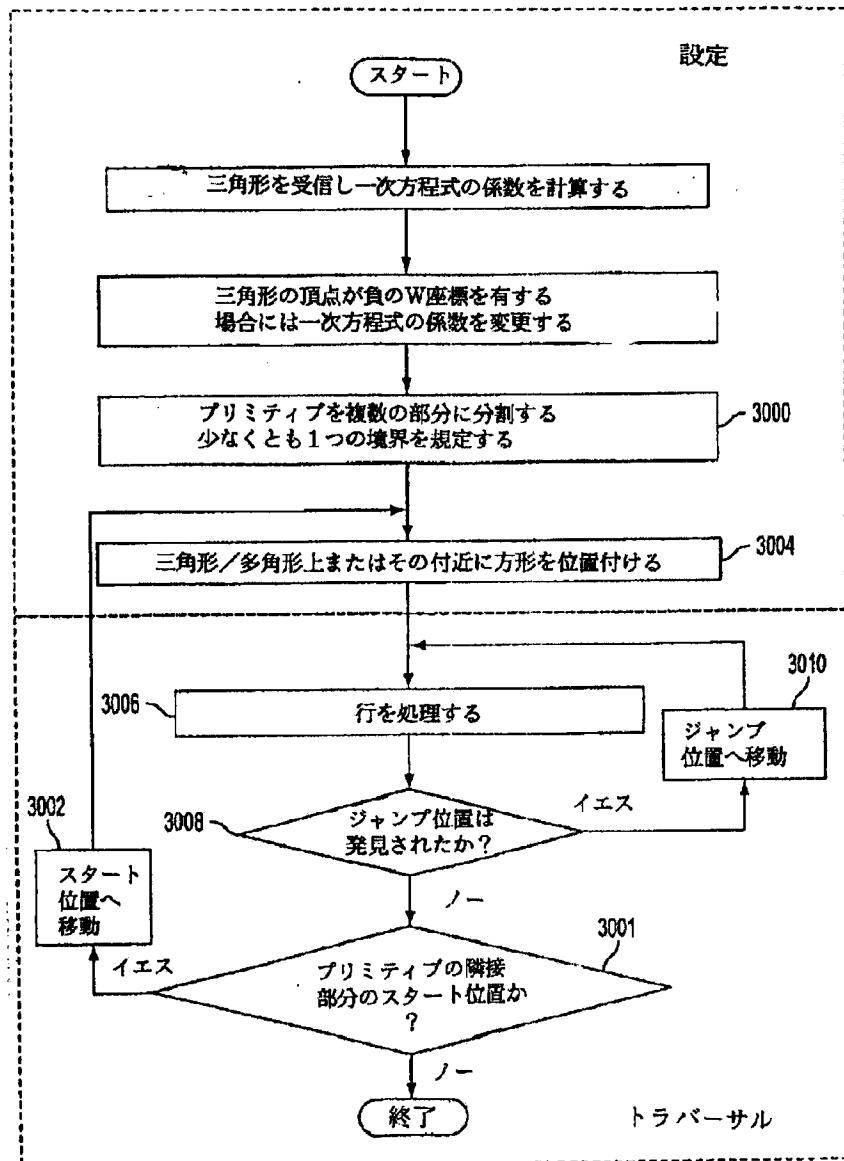
【図29】



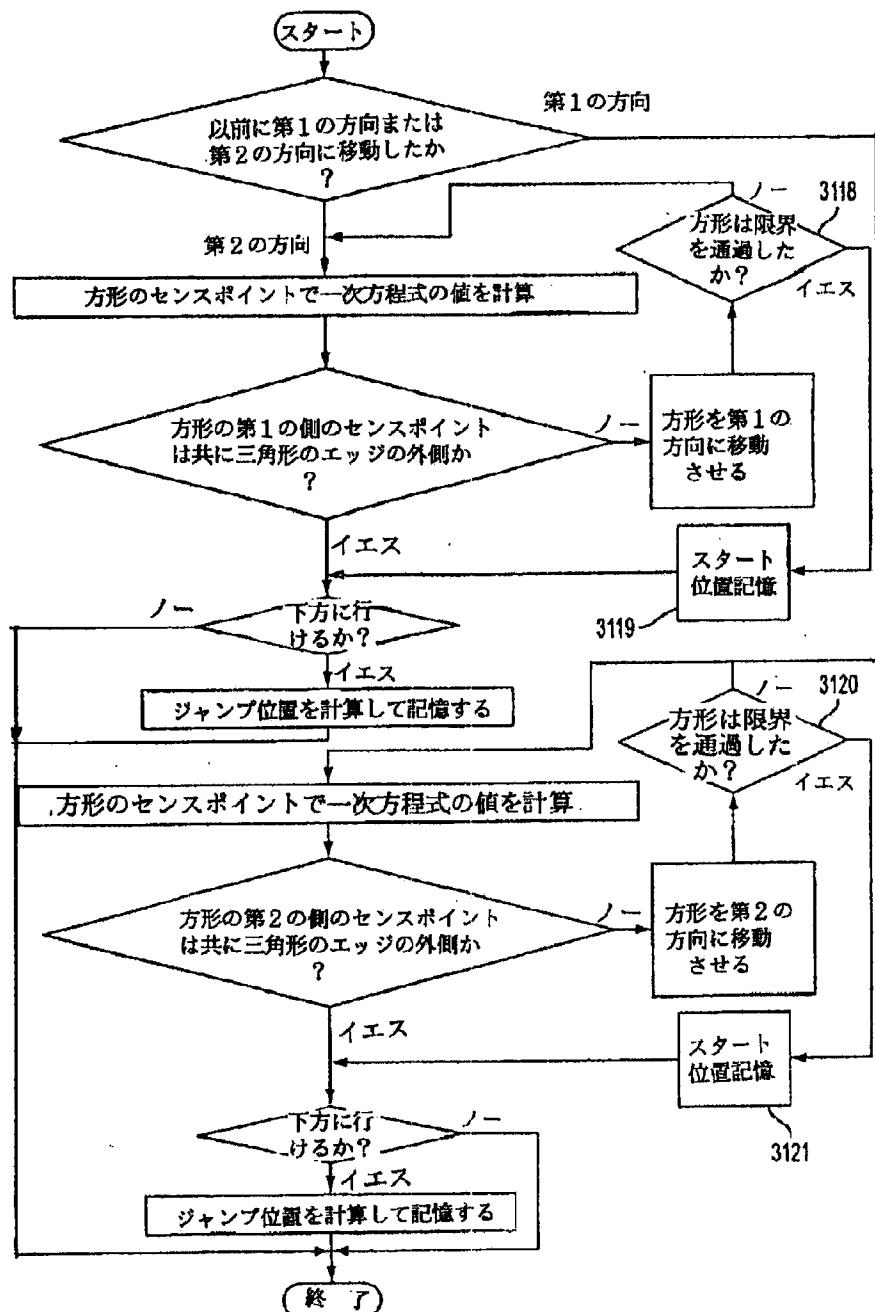
【図29A】

1		2850	3		
2			4		
5			7		
6			8		
10	9		13		
11	12		14		
16	15		19	20	
17	18				
24	23	22		21	

【四三〇】

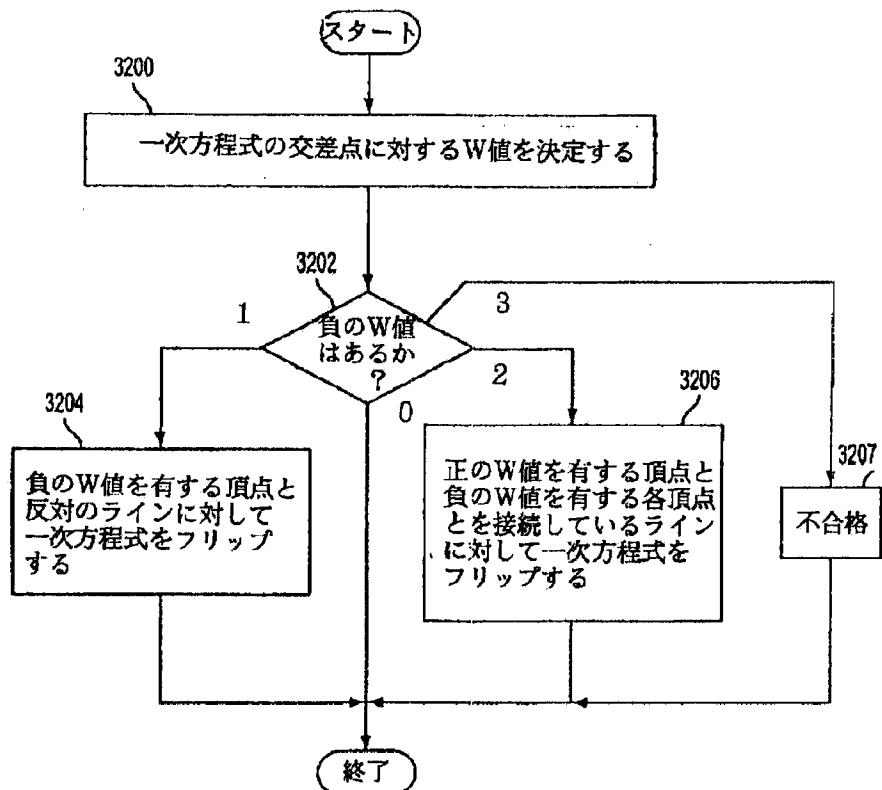


【図31】

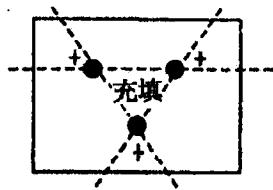


【☒ 3 1 A】

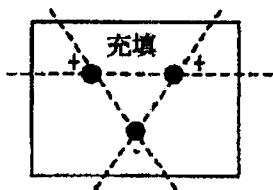
【図32】



【図32A】



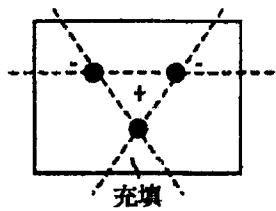
【図32B】



(113)

特表2003-515851

【図32C】



【手続補正書】特許協力条約第34条補正の翻訳文提出書

【提出日】平成14年3月8日(2002.3.8)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】 (a) 頂点データをベクトルの形態で受取り、ベクトル頂点データに関してベクトル演算を行うためのベクトル演算モジュールと、

(b) ベクトル演算モジュールからのスカラー頂点データをベクトル頂点データに変換するための、ベクトル演算モジュールに結合された変換モジュールと、

(c) ベクトル演算モジュールの出力を記憶して、その出力をベクトル演算モジュールにフィードバックするための、ベクトル演算モジュールに結合されたレジスタとを具備しているグラフィック処理中にスカラーおよびベクトル成分を処理するためのシステム。

【請求項2】 ベクトル演算モジュールは乗算器および加算器の少なくとも一方を含んでいる請求項1記載のシステム。

【請求項3】 ゼロレイテンシーはレジスタをバイパスすることによって達成される請求項1記載のシステム。

【請求項4】 レジスタは、ベクトル頂点データを発生するベクトル成分書き込みマスクを含んでいる請求項3記載のシステム。

【請求項5】 さらに、ベクトル演算モジュールの出力に関してスカラー演算を実行し、それによって頂点データをスカラーの形態でレンダリングするための適合されたスカラー演算モジュールを備えている請求項1記載のシステム。

【請求項6】 スカラー演算には、逆数または逆平方根演算が含まれる請求項5記載のシステム。

【請求項7】 (a) 頂点データをベクトルの形態で受取り、

(b) ベクトル頂点データに関してベクトル演算を行い、

(c) ベクトル演算から結果的に得られたスカラー頂点データをベクトル頂点データに変換し、

(d) ベクトル演算の出力を記憶し、

(e) ベクトル演算の記憶された出力に関して付加的なベクトル演算を行うステップを含んでいるグラフィック処理中にスカラーおよびベクトル成分を処理する方法。

【請求項8】 ベクトル演算は乗算または加算演算を含んでいる請求項7記載の方法。

【請求項9】 ベクトル演算はゼロレイテンシーでベクトル演算の出力に関して行われる請求項7記載の方法。

【請求項10】 ベクトル演算の出力はレジスタ装置に記憶され、ゼロレイテンシーはレジスタ装置をバイパスすることによって達成される請求項9記載の方法。

【請求項11】 ベクトル演算の出力に関してスカラー演算を実行し、それによって頂点データをスカラーの形態でレンダリングするステップをさらに含んでいる請求項7記載の方法。

【請求項12】 スカラー演算は逆数または逆平方根演算を含んでいる請求項11記載の方法。

【請求項13】 ベクトル演算の出力がベクトルの形態である場合、その出力からスカラー頂点データを抽出するステップをさらに含んでいる請求項11記載の方法。

【請求項14】 抽出はマルチプレクサによって行なわれる請求項13記載の方法。

【請求項15】 受取られた頂点データはマルチプレクサによって操作される請求項7記載の方法。

【請求項16】 (a) 頂点データをベクトルの形態で受取るためのコードセグメントと、

(b) ベクトル頂点データに関してベクトル演算を行うためのコードセグメントと、

(c) ベクトル演算から結果的に得られたスカラー頂点データをベクトル頂点データに変換するためのコードセグメントとを含み、

(d) ベクトル演算の出力を記憶し、

(e) ベクトル演算の記憶された出力に関して付加的なベクトル演算を行うグラフィック処理中にスカラーおよびベクトル成分を処理するためのコンピュータ読出し可能な媒体上に支持されているコンピュータプログラム。

【請求項17】 ベクトル演算は乗算または加算演算を含んでいる請求項16記載のコンピュータプログラム。

【請求項18】 ベクトル演算はゼロレイテンシーでベクトル演算の出力について行われる請求項16記載のコンピュータプログラム。

【請求項19】 ベクトル演算の出力はレジスタ装置に記憶され、ゼロレイテンシーはレジスタ装置をバイパスすることによって達成される請求項18記載のコンピュータプログラム。

【請求項20】 さらに、ベクトル演算の出力に関してスカラー演算を実行し、それによって頂点データをスカラーの形態でレンダリングするコードセグメントを含んでいる請求項16記載のコンピュータプログラム。

【請求項21】 スカラー演算は逆数または逆平方根演算を含んでいる請求項20記載のコンピュータプログラム。

【請求項22】 さらに、ベクトル演算の出力がベクトルの形態である場合、その出力からスカラー頂点データを抽出するコードセグメントを含んでいる請求項20記載のコンピュータプログラム。

【請求項23】 抽出はマルチプレクサによって行なわれる請求項22記載のコンピュータプログラム。

【請求項24】 (a) 頂点データをベクトルの形態で受取り、
(b) ベクトル頂点データに関してベクトル演算を行い、そのベクトル演算は乗算または加算演算を含み、

(c) ベクトル演算から結果的に得られたスカラー頂点データをベクトル頂点データに変換し、

(d) ベクトル演算の出力をレジスタ装置に記憶し、

- (e) ベクトル演算の記憶された出力に関して付加的なベクトル演算を行い、
- (f) ベクトル演算の記憶された出力に関してスカラー演算を実行し、それに
よって頂点データをスカラーの形態でレンダリングし、スカラー演算は逆数また
は逆平方根演算を含み、
- (g) ベクトル演算の記憶された出力がベクトルの形態である場合、その記憶
された出力からスカラー頂点データを抽出するステップを含んでいるグラフィッ
ク処理中にスカラーおよびベクトル成分を処理する方法。

【請求項25】 (a) 複数のマトリックス、複数の加重値および頂点データをバッファにおいて受取り、加重値はブレンディングした加重値を含んでおり

、
(b) 複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、
および加重値の少なくとも1つとの乗算により計算されたものであり、

(c) 付加的な処理のために積の和を出力するステップを含んでおり、

(d) 積の和は $i = 1 \cdots x$ に対して $\sum w_i * M_i * v$ として計算され、
ここで、 $v = \text{頂点データ}$ 、

$w = \text{加重値}$ 、

$M = \text{マトリックス}$ 、および

$x = \text{マトリックスの数}$

であるハードウェア構成グラフィックパイプラインにおけるグラフィック処理中
にブレンディング演算を行う方法。

【請求項26】 マトリックスはモデルビューマトリックスを含んでいる請
求項25記載の方法。

【請求項27】 付加的な処理には、表示するための積の和と合成マトリッ
クスとの乗算が含まれている請求項26記載の方法。

【請求項28】 付加的な処理には、ライティング動作が含まれている請求
項26記載の方法。

【請求項29】 マトリックスは逆マトリックスを含み、頂点データは正規
ベクトルを含んでいる請求項25記載の方法。

【請求項30】 (a) 複数のマトリックス、複数の加重値および頂点データ

タをバッファにおいて受取り、

- (b) 複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重値の少なくとも1つとの乗算により計算されたものであり、
- (c) 付加的な処理のために積の和を出力するステップを含んでおり、
- (d) ステップ(a)乃至(c)は、バッファの出力に結合された第1の入力を有し、頂点データを受取る乗算論理装置と、この乗算論理装置の出力に結合された第1の入力を有する演算論理装置と、この演算論理装置の出力に結合された入力を有するレジスタ装置と、ならびに乗算論理装置および演算論理装置に結合されたメモリとを備えた单一の集積回路によって行われ、メモリには、頂点データを処理するときのための複数の定数および変数が記憶されている、ハードウェア構成グラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行う方法。

【請求項31】 (a) 複数のマトリックス、複数の加重値および頂点データをバッファにおいて受取り、

- (b) 複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重値の少なくとも1つとの乗算により計算されたものであり、
- (c) 付加的な処理のために積の和を出力するステップを含んでおり、
- (d) ステップ(a)乃至(c)は、バッファの出力に結合された第1の入力を有する乗算論理装置と、この乗算論理装置の出力に結合された第1の入力を有する演算論理装置と、この演算論理装置の出力に結合された入力を有するレジスタ装置と、演算論理装置またはレジスタ装置の出力に結合された入力を備えた反転論理装置と、この反転論理装置の出力と乗算論理装置の第2の入力との間に結合された変換モジュールと、乗算論理装置および演算論理装置に結合されたメモリと、ならびに演算論理装置の出力に結合された出力コンバータとを備えた单一の集積回路によって行われ、反転論理装置は逆数または逆平方根演算を行い、変換モジュールはスカラー頂点データをベクトル調印データに変換するように構成されており、メモリには頂点データを処理するときに使用される複数の定数および変数が記憶されており、出力コンバータは処理された頂点データを出力するようにライティングモジュールに結合されている、ハードウェア構成グラフィック

パイプラインにおけるグラフィック処理中にブレンディング演算を行う方法。

【請求項32】 (a) 複数のマトリックス、ブレンディング加重値を含む複数の加重値、および頂点データを受取るためのバッファと、

(b) このバッファに結合された单一の集積回路とを備えており、この集積回路は複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重値の少なくとも1つとの乗算により計算されたものであり、

(c) 付加的な処理のために单一の集積回路から積の和が出力され、この積の和は $i = 1 \cdots x$ に対して $\sum w_i * M_i * v$ として計算され、

ここで、 v = 頂点データ、

w = 加重値、

M = マトリックス、および

x = マトリックスの数

であるグラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行うためのシステム。

【請求項33】 マトリックスはモデルビューマトリックスを含んでいる請求項32記載のシステム。

【請求項34】 付加的な処理には、表示するための積の和と合成マトリックスとの乗算が含まれている請求項33記載のシステム。

【請求項35】 付加的な処理には、ライティング演算が含まれている請求項33記載のシステム。

【請求項36】 マトリックスは逆マトリックスを含み、頂点データは正規ベクトルを含んでいる請求項32記載のシステム。

【請求項37】 (a) 複数のマトリックス、複数の加重値、および頂点データを受取るためのバッファと、

(b) このバッファに結合された单一の集積回路とを備えており、この集積回路は複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重値の少なくとも1つとの乗算により計算されたものであり、

(c) 付加的な処理のために单一の集積回路から積の和が出力され、单一の集積回路は、バッファの出力に結合された第1の入力を有し、頂点データを受取る

乗算論理装置と、この乗算論理装置の出力に結合された第1の入力を有する演算論理装置と、この演算論理装置の出力に結合された入力を有するレジスタ装置と、ならびに乗算論理装置および演算論理装置に結合されたメモリとを備えた单一の集積回路によって行われ、メモリには、頂点データを処理するときに使用される複数の定数および変数が記憶されている、グラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行うためのシステム。

【請求項38】 (a) 複数のマトリックスと、複数の加重値と、および頂点データとを受取るためのバッファと、

(b) このバッファに結合された单一の集積回路とを備えており、この集積回路は複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重値の少なくとも1つとの乗算により計算されたものであり、

(c) 積の和は付加的な処理のために单一の集積回路から出力され、单一の集積回路は、バッファの出力に結合された第1の入力を有する乗算論理装置と、この乗算論理装置の出力に結合された第1の入力を有する演算論理装置と、この演算論理装置の出力に結合された入力を有するレジスタ装置と、演算論理装置またはレジスタ装置の出力に結合された入力を備えた反転論理装置と、この反転論理装置の出力と乗算論理装置の第2の入力との間に結合された変換モジュールと、乗算論理装置および演算論理装置に結合されたメモリと、ならびに演算論理装置の出力に結合された出力コンバータとを備えており、反転論理装置は逆数または逆平方根演算を行い、変換モジュールはスカラー頂点データをベクトル調印データに変換するように構成されており、メモリには頂点データを処理するときに使用される複数の定数および変数が記憶されており、出力コンバータは処理された頂点データを出力するようにライティングモジュールに結合されている、グラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行うためのシステム。

【請求項39】 (a) 複数のマトリックスと、複数の加重値と、および頂点データとを受取り、

(b) ブレンディング演算を行うために複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重値の少なくとも1つとの乗算により

計算されたものであり、

(c) 頂点データを照明し、

(d) 上記の各ステップは単一の半導体プラットフォーム上で行われ、積の和は $i = 1 \cdots x$ に対して $\sum w_i * M_i * v$ として計算され、

ここで、 v = 頂点データ、

w = 加重値、

M = マトリックス、および

x = マトリックスの数

であるハードウェア構成グラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行う方法。

【請求項 40】 (a) 頂点データを受取るように構成された変換モジュールと、

(b) この変換モジュールに結合され、前記変換モジュールと同じ半導体プラットフォーム上に位置されたライティングモジュールとを備えており、

前記変換モジュールは、前記単一の半導体プラットフォーム上に位置されており、頂点データを第 1 の空間から第 2 の空間に変換し、さらにバッファおよびこのバッファに結合された論理装置を含んでおり、バッファが複数のマトリックス、複数の加重値および頂点データを受取り、論理装置がブレンディング演算を行うために複数の積の和を計算し、各積は頂点データと、マトリックスの 1 つと、および加重値の少なくとも 1 つとの乗算により計算されたものであり、

前記ライティングモジュールは、変換モジュールから受取られた頂点データに関するライティング演算を行い、積の和は $i = 1 \cdots x$ に対して $\sum w_i * M_i * v$ として計算され、

ここで、 v = 頂点データ、

w = 加重値、

M = マトリックス、および

x = マトリックスの数

であるグラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行うためのシステム。

【請求項41】 各加重値はマトリックスの1つに対応している請求項40記載のシステム。

【請求項42】 加重値の和は1である請求項40記載のシステム。

【請求項43】 頂点データはx値、y値、z値およびw値を含んでいる請求項40記載のシステム。

【請求項44】 (a) 複数のマトリックスと、ブレンディング加重値を含む複数の加重値と、および頂点データとを受取り、

(b) 頂点データに関してブレンディング演算を行うために複数の積の和を計算し、各積は頂点データと、マトリックスの1つと、および加重値の少なくとも1つとの乗算により計算されたものであり、

(c) 付加的な処理のために積の和を出力し、

(d) この積の和は $i = 1 \cdots x$ に対して $\sum w_i * I_i * n$ として計算され、ここで、v = 頂点データ、

n = 正規ベクトルを含む頂点データ、

w = 加重値、

I = 逆転置マトリックス

x = マトリックスの数

であるステップを含んでいるハードウェア構成グラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行う方法。

【請求項45】 (a) 複数のマトリックスと、ブレンディング加重値を含む複数の加重値と、および頂点データとを受取るためのバッファと、

(b) このバッファに結合されて頂点データに関してブレンディング演算を行うために複数の積の和を計算する单一の集積回路とを備えており、各積は頂点データと、マトリックスの1つと、および加重値の少なくとも1つとの乗算により計算されたものであり、

(c) 付加的な処理のために单一の集積回路から積の和が output され、この積の和は $i = 1 \cdots x$ に対して $\sum w_i * I_i * n$ として計算され、

ここで、v = 頂点データ、

n = 正規ベクトルを含む頂点データ、

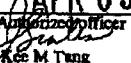
w = 加重値、

I = 逆転置マトリックス

x = マトリックスの数

であるグラフィックパイプラインにおけるグラフィック処理中にブレンディング演算を行うためのシステム。

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US00/33043																								
A. CLASSIFICATION OF SUBJECT MATTER IPC(7) : G06T 01/20 US CL : 345/505 According to International Patent Classification (IPC) or to both national classification and IPC																										
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) U.S. : 345/505, 501-503, 506-509, 512, 513, 515, 516, 522-524, 426, 427, 440-443																										
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched none																										
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WEST: scalar, vector, vertex																										
C. DOCUMENTS CONSIDERED TO BE RELEVANT <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Category *</th> <th style="width: 80%;">Citation of document, with indication, where appropriate, of the relevant passages</th> <th style="width: 10%;">Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>US 5,801,711 A (KOSS et al) 01 September 1998, Figures 1 and 3, col. 7, lines 1-20; col. 17, line 10 to col. 18, line 64.</td> <td>48-52, 55-59</td> </tr> <tr> <td>—</td> <td></td> <td>53-54, 60-61</td> </tr> <tr> <td>A</td> <td></td> <td>48-52, 55-59</td> </tr> <tr> <td>X, P</td> <td>US 6,057,855 A (BARKANS) 02 May 2000, figures 2 and 3, col. 5, line 54 through col. 6, line 37.</td> <td>48-52, 55-59</td> </tr> <tr> <td>X, P</td> <td>US 6,137,497A (STRUNK et al) 24 October 2000, figures 4 and 6, col. 12, 9-30; col. 11, lines 10-45.</td> <td>48-52, 55-59</td> </tr> <tr> <td>Y</td> <td>US 5,025,407 A (GULLEY et al) 18 June, 1991, col. 18, line 25 through col. 19, line 20.</td> <td>23-47</td> </tr> <tr> <td>Y</td> <td>US 5,22,202 A (KOYAMADA) 22 June 1993, abstract, col. 2, lines 15-33; col. 2, line 66 through col. 3, line 35.</td> <td>23-47</td> </tr> </tbody> </table>			Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	X	US 5,801,711 A (KOSS et al) 01 September 1998, Figures 1 and 3, col. 7, lines 1-20; col. 17, line 10 to col. 18, line 64.	48-52, 55-59	—		53-54, 60-61	A		48-52, 55-59	X, P	US 6,057,855 A (BARKANS) 02 May 2000, figures 2 and 3, col. 5, line 54 through col. 6, line 37.	48-52, 55-59	X, P	US 6,137,497A (STRUNK et al) 24 October 2000, figures 4 and 6, col. 12, 9-30; col. 11, lines 10-45.	48-52, 55-59	Y	US 5,025,407 A (GULLEY et al) 18 June, 1991, col. 18, line 25 through col. 19, line 20.	23-47	Y	US 5,22,202 A (KOYAMADA) 22 June 1993, abstract, col. 2, lines 15-33; col. 2, line 66 through col. 3, line 35.	23-47
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.																								
X	US 5,801,711 A (KOSS et al) 01 September 1998, Figures 1 and 3, col. 7, lines 1-20; col. 17, line 10 to col. 18, line 64.	48-52, 55-59																								
—		53-54, 60-61																								
A		48-52, 55-59																								
X, P	US 6,057,855 A (BARKANS) 02 May 2000, figures 2 and 3, col. 5, line 54 through col. 6, line 37.	48-52, 55-59																								
X, P	US 6,137,497A (STRUNK et al) 24 October 2000, figures 4 and 6, col. 12, 9-30; col. 11, lines 10-45.	48-52, 55-59																								
Y	US 5,025,407 A (GULLEY et al) 18 June, 1991, col. 18, line 25 through col. 19, line 20.	23-47																								
Y	US 5,22,202 A (KOYAMADA) 22 June 1993, abstract, col. 2, lines 15-33; col. 2, line 66 through col. 3, line 35.	23-47																								
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.																										
* Special categories of cited documents: "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent published on or after the international filing date "L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed																										
Date of the actual completion of the international search 23 January 2001 (23.01.2001)		Date of mailing of the international search report APR 05 2001																								
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230		Authorized officer  Lee M. Tung Telephone No. 703-305-4700																								

INTERNATIONAL SEARCH REPORT		International application No. PCT/US00/33043
Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet) This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:		
1. <input type="checkbox"/> Claim Nos.: because they relate to subject matter not required to be searched by this Authority, namely:		
2. <input type="checkbox"/> Claim Nos.: because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:		
3. <input type="checkbox"/> Claim Nos.: because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).		
Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet) This International Searching Authority found multiple inventions in this international application, as follows: Please See Continuation Sheet		
1. <input type="checkbox"/> As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims. 2. <input type="checkbox"/> As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee. 3. <input checked="" type="checkbox"/> As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.: 23-61 4. <input type="checkbox"/> No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:		
Remark on Protest <input type="checkbox"/> The additional search fees were accompanied by the applicant's protest. <input checked="" type="checkbox"/> No protest accompanied the payment of additional search fees.		

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US00/33043

BOX II. OBSERVATIONS WHERE UNITY OF INVENTION IS LACKING This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group I. Claims 1-22, drawn to a transform system for graphics processing.

Group II. Claims 23-47, drawn to a scalar and vector conversion module for graphics processing.

Group III. Claims 48-61, drawn to a method for performing blending operation in graphics processing system.

Group IV. Claims 61-69, drawn to a method for handling an inverse operation in a graphics processing module.

The inventions listed as Groups I-IV do not relate to a single general inventive concept under PCT Rule 13.1 because they lack the same or corresponding special technical features for the following reasons: the invention Groups I-IV each has separate technical features such as, a transform operation of Group I, a scalar and vector conversion of Group II, a method for performing blending operation of Group III and a method for handling an inverse operation of Group IV. Therefore, unity of invention is lacking.

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW

(72)発明者 カーク、デビッド

アメリカ合衆国、カリフォルニア州
94123 サン・フランシスコ、プロデリック
・ストリート 2965

(72)発明者 サベラ、パオロ

アメリカ合衆国、カリフォルニア州
94588 ブレザントン、カンバーランド・
ギャップ・コート 3457

Fターム(参考) 5B057 CA13 CA17 CB12 CB16 CC04
CH05 CH09
5B080 AA14 CA04 DA07 DA08 FA02
FA03